

PENGEMBANGAN SISTEM APLIKASI MANAJEMEN UNIT TRANSFUSI DARAH (UTD) PMI MALANG

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Tafarrara Irsa Shabrina

NIM: 145150200111098



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN SISTEM APLIKASI MANAJEMEN UNIT
TRANSFUSI DARAH (UTD) PMI MALANG

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Tafarrara Irsa Shabrina
NIM: 145150200111098

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018

Telah diperiksa dan disetujui oleh:

Pembimbing I



Nurudin Santoso, S.T, M.T
NIP: 19740916 200012 1 001

Pembimbing II



Bayu Priyambadha, S.Kom, M.Kom
NIP: 19820909 200812 1 004

Mengetahui

Ketua Jurusan Teknik Informatika



Iri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Agustus 2018



Tafarrara Irsa Shabrina

NIM: 145150200111098

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD) PMI Malang” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih banyak kepada:

1. Bapak Muhammad Samsudin Nasution dan Ibu Irma Rosyida selaku orang tua penulis, Muhammad Ramsa Irsa Ukasa, Shakyla Irsa Faradisa, dan Muhammad Birly Irsa Bilalbir selaku adik penulis, dan seluruh keluarga besar atas segala perhatian, kasih sayang, penyemangat, dan kesabaran hati dalam mendidik dan menunggu skripsi penulis terselesaikan, serta senantiasa memberikan doa dan semangat demi terselesainya skripsi ini,
2. Bapak Nurudin Santoso, S.T., M.T. dan Bapak Bayu Priyambadha, S.Kom., M.Kom. selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika,
4. Anton Firdaus, Yudistira Sugandi, Jawara Wahyu, Richa Amalia dan Elke Cahya yang sudah banyak membantu dan mengajarkan banyak ilmu kepada penulis,
5. Ditya Enandini, Sema Nabilah, Fendra Gunawan, Yudha Yusi, Andriko Hedi, dan Sastra Ginata, yang senantiasa memberikan fasilitas, memberi semangat, dan menghibur penulis,
6. Clara Pusparani, Rony Hendiarto, dan Nur Afdaliyah yang senantiasa menjadi teman seperjuangan disaat menjalankan skripsi,
7. Nur Adli Ari Darmawand, Afifahya Alif Rizyomi, dan Evelina Meisya yang senantiasa menjadi wadah segala keluh kesah penulis dan pemberi semangat,
8. Teman-teman Kopma Squad, BE’M nunu ditya, Bismit, Nol Derajat, dan penunggu antrian bimbingan yang senantiasa menjadi motivasi, menemani, dan menghibur penulis,
9. Semua pihak yang tidak dapat disebutkan satu persatu, terima kasih banyak atas bantuan, semangat serta doa yang senantiasa diberikan kepada penulis.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari kata sempurna, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang membaca dan menggunakannya.

Malang, 1 Agustus 2018

Penulis

tafarrarashabrina@gmail.com

ABSTRAK

Tafarrara Irsa Shabrina, Pengembangan Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD) PMI Malang

Pembimbing: Nurudin Santoso, S.T., M.T. dan Bayu Priyambadha, S.Kom., M.Kom.

Unit Transfusi Darah (UTD) merupakan suatu wadah pelayanan kesehatan untuk masyarakat dengan tujuan untuk membantu masyarakat terkait dengan darah. Untuk membantu masyarakat dan rumah sakit dalam mengatasi permasalahan, seperti permohonan permintaan darah dari rumah sakit menggunakan cara yang berbeda antara UTD cabang dan bukan UTD cabang, terdapat tidak jelasnya jumlah stok darah yang ada di UTD pusat Kota Malang, dan tidak adanya transparansi proses manajemen darah dari awal permohonan permintaan darah hingga darah diterima, serta donor darah dilakukan dengan cara manual. Solusi yang diberikan adalah memanajemen data permintaan darah, menyamaratakan atau melihat keurgensian dari permintaan, memberikan informasi stok darah secara *real time*, adanya transparansi pengiriman darah hingga darah sampai ke rumah sakit, meminimalisirkan waktu dalam proses pengajuan donor darah dan mendapatkan informasi akurat mengenai permohonan darah yang stok darah tersebut sedang tidak terdapat pada UTD. Dalam sistem akan menerapkan *waterfall model* untuk proses pengembangannya. Hasil dari pengujian unit dan pengujian validasi menghasilkan nilai 100% valid. Pada pengujian kompatibilitas dihasilkan sistem dapat berjalan pada 9 jenis peramban.

Kata kunci: sistem manajemen UTD, rekayasa perangkat lunak

ABSTRACT

Tafarrara Irsa Shabrina, Pengembangan Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD) PMI Malang

Advisor: Nurudin Santoso, S.T., M.T. dan Bayu Priyambadha, S.Kom., M.Kom.

Blood transfusion unit is a container of medical services to the community in order to help the community associated with something about blood. To help the community and hospitals in resolve the problem, such as a blood request from hospitals using different ways between UTD branch and who do not have UTD branch, there are no details of the number of existing blood stock in Malang Central UTD, and the lack of transparency of the management process from the initial application blood request until it is received, and blood donation is done manually. The given solution is create a data blood request, leveler or see an urgency of the request, providing information about blood stock in real time, the existence of a blood delivery to the transparency of the blood up to the hospital, minimize time in the process of filing blood donation and get an accurate information about blood request if a blood stock are not available in UTD. The system will apply the waterfall model for the process of development. Results of unit testing, integration testing and validation testing yields a value of 100% valid. The compatibility testing resulted that the system can run on browser type.

Keywords: Blood transfusion unit system, software engineering

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Manajemen	6
2.1.1 Manajemen Unit Transfusi Darah (UTD).....	6
2.2 <i>Software Development Life Cycle (SDLC)</i>	9
2.2.1 <i>Model Waterfall</i>	9
2.3 <i>Framework Codeigniter</i>	10
2.4 <i>Unified Modeling Language (UML)</i>	11
2.4.1 <i>Use Case Diagram</i>	12
2.4.2 <i>Sequence Diagram</i>	12
2.4.3 <i>Class Diagram</i>	12
2.5 Pengujian Perangkat Lunak.....	13
2.5.2 <i>Whitebox Testing</i>	13
2.5.3 <i>Blackbox Testing</i>	13
BAB 3 METODOLOGI PENELITIAN	15

3.1 Tipe Penelitian	15
3.2 Strategi dan Rancangan Penelitian	15
3.2.1 Studi Kepustakaan.....	16
3.2.2 Pengumpulan Data.....	16
3.2.3 Analisis Kebutuhan.....	16
3.2.4 Perancangan Sistem.....	17
3.2.5 Implementasi Sistem.....	18
3.2.6 Pengujian Sistem	18
3.2.7 Kesimpulan dan Saran.....	18
BAB 4 ANALISIS KEBUTUHAN	20
4.1 Elisitasi Kebutuhan.....	20
4.2 Deskripsi Sistem	20
4.3 Identifikasi Aktor.....	21
4.4 Spesifikasi Kebutuhan	22
4.5 Pemodelan Kebutuhan Sistem.....	40
4.5.1 <i>Use Case Diagram</i>	40
4.5.2 <i>Use Case Scenario</i>	42
BAB 5 PERANCANGAN DAN IMPLEMENTASI	73
5.1 Perancangan	73
5.1.1 <i>Sequence Diagram</i>	73
5.1.2 Perancangan Basis Data	76
5.1.3 <i>Class Diagram</i>	77
5.1.4 Perancangan Algoritme.....	79
5.1.5 Perancangan Antarmuka.....	82
5.2 Implementasi	84
5.2.1 Spesifikasi Sistem	84
5.2.2 Implementasi Algoritme.....	85
5.2.3 Implementasi Antarmuka	92
BAB 6 PENGUJIAN	94
6.1 Pengujian Unit.....	94
6.1.1 Pengujian Unit Menambah Alur Pengiriman Darah	94
6.1.2 Pengujian Unit Melihat Informasi Butuh Darah.....	97

6.1.3 Pengujian Unit Menambah Permintaan Darah.....	101
6.2 Pengujian Validasi	103
6.3 Pengujian <i>Compatibility</i>	150
BAB 7 PENUTUP	152
7.1 Kesimpulan.....	152
7.2 Saran	153
DAFTAR PUSTAKA.....	154
LAMPIRAN A DOKUMENTASI WAWANCARA	156
LAMPIRAN B FORMULIR PENGAJUAN DONOR DARAH.....	160
LAMPIRAN C FORMULIR PERMINTAAN DARAH.....	164



DAFTAR TABEL

Tabel 4.1 Identifikasi Aktor	21
Tabel 4.2 Kebutuhan Fungsional.....	22
Tabel 4.3 Kebutuhan Non-Fungsional.....	40
Tabel 4.4 <i>Use Case Scenario</i> Pendaftaran.....	42
Tabel 4.5 <i>Use Case Scenario Login</i>	42
Tabel 4.6 <i>Use Case Scenario Logout</i>	43
Tabel 4.7 <i>Use Case Scenario</i> Melihat Stok Darah	43
Tabel 4.8 <i>Use Case Scenario</i> Melihat Alur Donor	43
Tabel 4.9 <i>Use Case Scenario</i> Melihat Informasi Setelah Donor.....	44
Tabel 4.10 <i>Use Case Scenario</i> Menambah Pesan Hubungi.....	44
Tabel 4.11 <i>Use Case Scenario</i> Melihat Daftar Pengguna.....	45
Tabel 4.12 <i>Use Case Scenario</i> Menambah Stok Darah	45
Tabel 4.13 <i>Use Case Scenario</i> Menambah Pengguna	46
Tabel 4.14 <i>Use Case Scenario</i> Mengubah Data Pengguna.....	46
Tabel 4.15 <i>Use Case Scenario</i> Menghapus Pengguna.....	47
Tabel 4.16 <i>Use Case Scenario</i> Melihat Daftar Permintaan Darah	47
Tabel 4.17 <i>Use Case Scenario</i> Menambah Permintaan Darah	48
Tabel 4.18 <i>Use Case Scenario</i> Menambah Alur Pengiriman Darah	48
Tabel 4.19 <i>Use Case Scenario</i> Melihat Detail Permintaan Darah	49
Tabel 4.20 <i>Use Case Scenario</i> Mengubah Data Permintaan Darah	49
Tabel 4.21 <i>Use Case Scenario</i> Melihat Daftar Data Petugas.....	50
Tabel 4.22 <i>Use Case Scenario</i> Mengubah Data Petugas.....	50
Tabel 4.23 <i>Use Case Scenario</i> Menghapus Data Petugas	51
Tabel 4.24 <i>Use Case Scenario</i> Melihat Daftar Pengajuan Donor	51
Tabel 4.25 <i>Use Case Scenario</i> Menambah Pengajuan Donor	52
Tabel 4.26 <i>Use Case Scenario</i> Melihat Detail Pengajuan Donor.....	52
Tabel 4.27 <i>Use Case Scenario</i> Menghapus Pengajuan Donor.....	53
Tabel 4.28 <i>Use Case Scenario</i> Melihat Daftar Data Pendonor.....	53
Tabel 4.29 <i>Use Case Scenario</i> Melihat Detail Data Pendonor	54
Tabel 4.30 <i>Use Case Scenario</i> Mengubah Data Pendonor.....	54

Tabel 4.31 <i>Use Case Scenario</i> Menghapus Data Pendonor	55
Tabel 4.32 <i>Use Case Scenario</i> Melihat Daftar Riwayat Donor Pendonor	55
Tabel 4.33 <i>Use Case Scenario</i> Menambah Riwayat Donor	55
Tabel 4.34 <i>Use Case Scenario</i> Melihat Detail Riwayat Donor	56
Tabel 4.35 <i>Use Case Scenario</i> Mengubah Data Riwayat Donor	56
Tabel 4.36 <i>Use Case Scenario</i> Menghapus Riwayat Donor	57
Tabel 4.37 <i>Use Case Scenario</i> Melihat Daftar Data Butuh Darah	57
Tabel 4.38 <i>Use Case Scenario</i> Menambah Data Butuh Darah	58
Tabel 4.39 <i>Use Case Scenario</i> Melihat Detail Butuh Darah	58
Tabel 4.40 <i>Use Case Scenario</i> Mengubah Data Butuh Darah	59
Tabel 4.41 <i>Use Case Scenario</i> Menghapus Data Butuh Darah	59
Tabel 4.42 <i>Use Case Scenario</i> Melihat Pemberitahuan	60
Tabel 4.43 <i>Use Case Scenario</i> Melihat Syarat Donor	60
Tabel 4.44 <i>Use Case Scenario</i> Menambah Pengajuan Donor	60
Tabel 4.45 <i>Use Case Scenario</i> Mencetak Pengajuan Donor	61
Tabel 4.46 <i>Use Case Scenario</i> Melihat Informasi Butuh Darah	61
Tabel 4.47 <i>Use Case Scenario</i> Melihat Profil Pendonor	62
Tabel 4.48 <i>Use Case Scenario</i> Mengubah Profil Pendonor	63
Tabel 4.49 <i>Use Case Scenario</i> Melihat Penghargaan Donor	63
Tabel 4.50 <i>Use Case Scenario</i> Melihat Riwayat Donor	64
Tabel 4.51 <i>Use Case Scenario</i> Menambah Pengajuan Permintaan Darah	64
Tabel 4.52 <i>Use Case Scenario</i> Melihat Daftar Pengajuan Permintaan Darah	65
Tabel 4.53 <i>Use Case Scenario</i> Melihat Detail Pengajuan Permintaan Darah	65
Tabel 4.54 <i>Use Case Scenario</i> Mencetak Pengajuan Permintaan Darah	66
Tabel 4.55 <i>Use Case Scenario</i> Melihat Daftar Permintaan Disetujui	66
Tabel 4.56 <i>Use Case Scenario</i> Melihat Detail Permintaan Disetujui	66
Tabel 4.57 <i>Use Case Scenario</i> Melihat Daftar Darah Diterima	67
Tabel 4.58 <i>Use Case Scenario</i> Melihat Detail Darah Diterima	67
Tabel 4.59 <i>Use Case Scenario</i> Melihat Daftar Permintaan Selesai	68
Tabel 4.60 <i>Use Case Scenario</i> Melihat Detail Permintaan Selesai	68
Tabel 4.61 <i>Use Case Scenario</i> Mencetak Permintaan Selesai	69
Tabel 4.62 <i>Use Case Scenario</i> Melihat Daftar Permintaan Dibatalkan	69

Tabel 4.63 <i>Use Case Scenario</i> Melihat Detail Permintaan Dibatalkan.....	69
Tabel 4.64 <i>Use Case Scenario</i> Mencetak Permintaan Dibatalkan	70
Tabel 4.65 <i>Use Case Scenario</i> Melihat Syarat Permintaan Darah	70
Tabel 4.66 <i>Use Case Scenario</i> Melihat Informasi Alur Permintaan	71
Tabel 4.67 <i>Use Case Scenario</i> Melihat Profil Petugas Rumah Sakit.....	71
Tabel 4.68 <i>Use Case Scenario</i> Mengubah Profil Petugas Rumah Sakit.....	72
Tabel 5.1 Perancangan Algoritme Menambah Alur Pengiriman Darah.....	79
Tabel 5.2 Perancangan <i>Algoritme</i> Melihat Informasi Butuh Darah.....	80
Tabel 5.3 Perancangan <i>Algoritme</i> Menambah Permintaan Darah.....	81
Tabel 5.4 Spesifikasi Perangkat Keras	85
Tabel 5.5 Spesifikasi Perangkat Keras	85
Tabel 5.6 Implementas <i>Algoritme</i> Menambah Alur Pengiriman Darah	86
Tabel 5.7 Implementas <i>Algoritme</i> Menambah Alur Pengiriman Darah	88
Tabel 5.8 Implementas <i>Algoritme</i> Menambah Alur Pengiriman Darah	89
Tabel 6.1 Pembentukan Node Algoritme fungsi add_alurpengiriman().....	94
Tabel 6.2 Kasus Uji Menambah Alur Pengiriman Darah	96
Tabel 6.3 Perancangan <i>Algoritme</i> Melihat Informasi Butuh Darah.....	97
Tabel 6.4 Kasus Uji Melihat Informasi Butuh Darah	99
Tabel 6.5 Perancangan <i>Algoritme</i> Menambah Permintaan Darah.....	101
Tabel 6.6 Kasus Uji Menambah Permintaan Darah	103
Tabel 6.7 Kasus Uji Pendaftaran	104
Tabel 6.8 Kasus Uji Pendaftaran	104
Tabel 6.9 Kasus Uji Pendaftaran	105
Tabel 6.10 Kasus Uji Login.....	105
Tabel 6.11 Kasus Uji Login.....	105
Tabel 6.12 Kasus Uji Login.....	106
Tabel 6.13 Kasus Uji Logout	106
Tabel 6.14 Kasus Uji Melihat Stok Darah	106
Tabel 6.15 Kasus Uji Melihat Alur Donor	107
Tabel 6.16 Kasus Uji Melihat Informasi Setelah Donor	107
Tabel 6.17 Kasus Uji Menambah Pesan Hubungi.....	107
Tabel 6.18 Kasus Uji Menambah Pesan Hubungi.....	108

Tabel 6.19 Kasus Uji Melihat Daftar Pengguna	108
Tabel 6.43 Kasus Uji Menambah Stok Darah	108
Tabel 6.44 Kasus Uji Menghapus Permintaan Darah	109
Tabel 6.20 Kasus Uji Menambah Pengguna	109
Tabel 6.21 Kasus Uji Menambah Pengguna	109
Tabel 6.22 Kasus Uji Menambah Pengguna	110
Tabel 6.23 Kasus Uji Mengubah Pengguna	110
Tabel 6.24 Kasus Uji Mengubah Pengguna	111
Tabel 6.25 Kasus Uji Mengubah Pengguna	111
Tabel 6.26 Kasus Uji Menghapus Pengguna	111
Tabel 6.27 Kasus Uji Menghapus Pengguna	112
Tabel 6.28 Kasus Uji Melihat Daftar Permintaan Darah	112
Tabel 6.29 Kasus Uji Menambah Permintaan Darah	112
Tabel 6.30 Kasus Uji Menambah Permintaan Darah	113
Tabel 6.31 Kasus Uji Menambah Permintaan Darah	113
Tabel 6.32 Kasus Uji Menambah Permintaan Darah	114
Tabel 6.33 Kasus Uji Menambah Alur Pengiriman Darah	114
Tabel 6.34 Kasus Uji Menambah Menambah Alur Pengiriman Darah	115
Tabel 6.35 Kasus Uji Menambah Menambah Alur Pengiriman Darah	115
Tabel 6.36 Kasus Uji Menambah Menambah Alur Pengiriman Darah	116
Tabel 6.37 Kasus Uji Menambah Menambah Alur Pengiriman Darah	116
Tabel 6.38 Kasus Uji Menambah Menambah Alur Pengiriman Darah	116
Tabel 6.39 Kasus Uji Melihat Detail Permintaan Darah	117
Tabel 6.40 Kasus Uji Mengubah Data Permintaan Darah	117
Tabel 6.41 Kasus Uji Mengubah Data Permintaan Darah	118
Tabel 6.42 Kasus Uji Mengubah Data Permintaan Darah	118
Tabel 6.45 Kasus Uji Melihat Daftar Data Petugas	118
Tabel 6.46 Kasus Uji Mengubah Data Data Petugas	119
Tabel 6.47 Kasus Uji Mengubah Data Data Petugas	119
Tabel 6.48 Kasus Uji Mengubah Data Data Petugas	119
Tabel 6.49 Kasus Uji Menghapus Data Petugas	120
Tabel 6.50 Kasus Uji Menghapus Data Petugas	120

Tabel 6.51 Kasus Uji Melihat Daftar Pengajuan Donor	121
Tabel 6.52 Kasus Uji Menambah Pengajuan Donor	121
Tabel 6.53 Kasus Uji Menambah Pengajuan Donor	121
Tabel 6.54 Kasus Uji Menambah Pengajuan Donor	122
Tabel 6.55 Kasus Uji Melihat Detail Pengajuan Donor	122
Tabel 6.56 Kasus Uji Menghapus Pengajuan Donor	123
Tabel 6.57 Kasus Uji Menghapus Pengajuan Donor	123
Tabel 6.58 Kasus Uji Melihat Daftar Data Pendonor	123
Tabel 6.59 Kasus Uji Melihat Detail Data Pendonor	124
Tabel 6.60 Kasus Uji Mengubah Data Data Pendonor	124
Tabel 6.61 Kasus Uji Mengubah Data Data Pendonor	125
Tabel 6.62 Kasus Uji Mengubah Data Data Pendonor	125
Tabel 6.63 Kasus Uji Menghapus Data Pendonor	125
Tabel 6.64 Kasus Uji Menghapus Data Pendonor	126
Tabel 6.65 Kasus Uji Melihat Daftar Riwayat Donor Pendonor	126
Tabel 6.66 Kasus Uji Menambah Riwayat Donor	126
Tabel 6.67 Kasus Uji Menambah Riwayat Donor	127
Tabel 6.68 Kasus Uji Menambah Riwayat Donor Pendonor	127
Tabel 6.69 Kasus Uji Melihat Detail Riwayat Donor	128
Tabel 6.70 Kasus Uji Mengubah Data Riwayat Donor	128
Tabel 6.71 Kasus Uji Mengubah Data Riwayat Donor Pendonor	129
Tabel 6.72 Kasus Uji Mengubah Data Riwayat Donor Pendonor	129
Tabel 6.73 Kasus Uji Menghapus Riwayat Donor	129
Tabel 6.74 Kasus Uji Menghapus Riwayat Donor	130
Tabel 6.75 Kasus Uji Melihat Daftar Data Butuh Darah	130
Tabel 6.76 Kasus Uji Menambah Data Butuh darah	130
Tabel 6.77 Kasus Uji Menambah Data Butuh darah	131
Tabel 6.78 Kasus Uji Menambah Data Butuh darah	131
Tabel 6.79 Kasus Uji Melihat Detail Butuh darah	132
Tabel 6.80 Kasus Uji Mengubah Data Butuh darah	132
Tabel 6.81 Kasus Uji Mengubah Data Butuh darah	132
Tabel 6.82 Kasus Uji Mengubah Data Butuh darah	133

Tabel 6.83 Kasus Uji Menghapus Butuh darah	133
Tabel 6.84 Kasus Uji Menghapus Butuh darah	133
Tabel 6.85 Kasus Uji Melihat Daftar Data Butuh Darah	134
Tabel 6.86 Kasus Uji Melihat Syarat Donor	134
Tabel 6.87 Kasus Uji Menambah Pengajuan Donor	134
Tabel 6.88 Kasus Uji Menambah Data Butuh darah	135
Tabel 6.89 Kasus Uji Menambah Data Butuh darah	136
Tabel 6.90 Kasus Uji Mencetak Pengajuan Donor	136
Tabel 6.91 Kasus Uji Melihat Informasi Butuh Darah	136
Tabel 6.92 Kasus Uji Melihat Informasi Butuh Darah	137
Tabel 6.93 Kasus Uji Melihat Informasi Butuh Darah	137
Tabel 6.94 Kasus Uji Melihat Informasi Butuh Darah	137
Tabel 6.95 Kasus Uji Melihat Informasi Butuh Darah	138
Tabel 6.96 Kasus Uji Melihat Informasi Butuh Darah	138
Tabel 6.97 Kasus Uji Melihat Informasi Butuh Darah	138
Tabel 6.98 Kasus Uji Melihat Informasi Butuh Darah	139
Tabel 6.99 Kasus Uji Melihat Informasi Butuh Darah	139
Tabel 6.100 Kasus Uji Melihat Profil Pendoror	139
Tabel 6.101 Kasus Uji Mengubah Profil Pendoror	140
Tabel 6.102 Kasus Uji Mengubah Profil Pendoror	140
Tabel 6.103 Kasus Uji Mengubah Profil Pendoror	140
Tabel 6.104 Kasus Uji Melihat Penghargaan Donor	141
Tabel 6.105 Kasus Uji Melihat Riwayat Donor	141
Tabel 6.106 Kasus Uji Menambah Pengajuan permintaan darah.....	141
Tabel 6.107 Kasus Uji Menambah Pengajuan permintaan darah.....	142
Tabel 6.108 Kasus Uji Menambah Pengajuan permintaan darah.....	142
Tabel 6.109 Kasus Uji Menambah Pengajuan pengajuan permintaan darah.....	143
Tabel 6.110 Kasus Uji Melihat Daftar Pengajuan Permintaan Darah	143
Tabel 6.111 Kasus Uji Melihat Detail Pengajuan Permintaan Darah	144
Tabel 6.112 Kasus Uji Mencetak Pengajuan Permintaan Darah.....	144
Tabel 6.113 Kasus Uji Melihat Daftar Permintaan disetujui	145
Tabel 6.114 Kasus Uji Melihat Detail Permintaan disetujui.....	145

Tabel 6.115 Kasus Uji Melihat Daftar Darah diterima	145
Tabel 6.116 Kasus Uji Melihat Detail Darah diterima	146
Tabel 6.117 Kasus Uji Melihat Daftar Permintaan Selesai	146
Tabel 6.118 Kasus Uji Melihat Detail Permintaan Selesai	147
Tabel 6.119 Kasus Uji Mencetak Permintaan Selesai	147
Tabel 6.120 Kasus Uji Melihat Daftar Permintaan Dibatalkan	147
Tabel 6.121 Kasus Uji Melihat Detail Permintaan Dibatalkan	148
Tabel 6.122 Kasus Uji Mencetak Permintaan Dibatalkan	148
Tabel 6.123 Kasus Uji Melihat Syarat Permintaan Darah	148
Tabel 6.124 Kasus Uji Melihat Informasi Alur Permintaan	149
Tabel 6.125 Kasus Uji Melihat Profil Petugas Rumah Sakit	149
Tabel 6.126 Kasus Uji Mengubah Profil Petugas Rumah Sakit	149
Tabel 6.127 Kasus Uji Mengubah Profil Petugas Rumah Sakit	150
Tabel 6.128 Kasus Uji Mengubah Profil Petugas Rumah Sakit	150



DAFTAR GAMBAR

Gambar 2.1 Bagan Alur Donor Darah Unit Transfusi Darah (UTD) PMI Malang	7
Gambar 2.2 Bagan Alur Permintaan Darah Unit Transfusi Darah (UTD) PMI Malang	8
Gambar 2.3 Model <i>waterfall</i>	9
Gambar 3.1 <i>Diagram</i> Alir Metode Penelitian	15
Gambar 4.1 <i>Use Case Diagram</i>	41
Gambar 5.1 <i>Sequence Diagram</i> Menambah Alur Pengiriman Darah	74
Gambar 5.2 <i>Sequence Diagram</i> Melihat informasi Butuh Darah	75
Gambar 5.3 <i>Sequence Diagram</i> Menambah Permintaan Darah	76
Gambar 5.4 <i>Physical Data Model</i>	77
Gambar 5.5 Class Diagram	78
Gambar 5.6 Perancangan Antarmuka Menambah Alur Pengiriman Darah	82
Gambar 5.7 Perancangan Antarmuka Melihat Informasi Butuh Darah	83
Gambar 5.8 Perancangan antarmuka menambah permohonan donor	84
Gambar 5.9 Implementasi Antarmuka Menambah Alur Pengiriman Darah	92
Gambar 5.10 Implementasi Antarmuka Melihat Informasi Butuh Darah	93
Gambar 5.11 Implementasi Antarmuka Menambah Permintaan Darah	93
Gambar 6.1 Flowgraph Algoritme fungsi <i>add_alurpengiriman()</i>	95
Gambar 6.2 Flowgraph Algoritme fungsi <i>butuh()</i>	98
Gambar 6.3 Flowgraph Algoritme fungsi <i>add_permintaan()</i>	102
Gambar 6.4 Hasil Pengujian <i>Compatibility</i> Aplikasi	151
Gambar 6.5 Kategori <i>compatibility issues</i> pada <i>sortSite</i>	151

BAB 1 PENDAHULUAN

1.1 Latar belakang

Darah adalah bagian terpenting untuk kehidupan, yang merupakan kombinasi dari plasma dan sel-sel yang beredar melalui seluruh tubuh. Fungsi darah sendiri adalah untuk mengedarkan sari-sari makanan, oksigen serta zat-zat lain, sebagai pengantar antara bahan-bahan kimia yang dihasilkan oleh metabolisme tubuh, dan juga untuk menghindari tubuh dari virus atau bakteri yang akan menyerang (Widodo, 2012). Fungsi lainnya yaitu untuk melawan interaksi, mengatur suhu tubuh serta mengkoordinasikan aktifitas jaringan tubuh (Mader, 2012). Sel darah merah merupakan jenis sel terbanyak dalam darah yang dapat menjadi penentu jenis golongan darah seseorang. Apabila seseorang kehilangan darah dalam jumlah yang besar, baik disebabkan oleh kecelakaan, prosedur bedah medis, atau hal lainnya, maka tubuh manusia akan merasa terganggu akibat sulitnya dalam menghasilkan sel darah merah baru yang berfungsi untuk menggantikan darah yang telah banyak keluar tersebut. Ketika jumlah volume darah yang keluar melebihi volume darah yang berada dalam tubuh. Maka untuk menghindari kekurangan sel darah merah tersebut adalah dengan melakukan transfusi darah, dengan tujuan untuk mengganti darah yang telah banyak keluar.

Unit Transfusi Darah (UTD) merupakan suatu wadah pelayanan kesehatan untuk masyarakat yaitu dengan merencanakan, mengarahkan, dan melestarikan pendonor darah, dan juga menyediakan darah, mendistribusikan darah, serta melakukan tindakan medis berupa pemberian darah kepada pasien yang membutuhkan dengan tujuan untuk proses penyembuhan penyakit. Selanjutnya setiap Rumah Sakit memiliki UTD cabang masing-masing dengan tujuan memudahkan penyaluran darah kepada setiap bank darah yang terdapat pada setiap Rumah Sakit (PMK RI, 2015). UTD cabang tersebut dapat disebut juga sebagai bank darah rumah sakit, karena berinteraksi langsung dengan UTD pusat. Namun, tidak semua rumah sakit memiliki UTD cabang yang langsung terhubung kepada UTD pusat Kota Malang.

Dengan keterbatasan UTD cabang tersebut, maka yang terjadi adalah alur permohonan permintaan darah akan berbeda antara rumah sakit yang memiliki UTD cabang dengan yang tidak memiliki. Untuk permohonan permintaan darah UTD cabang setiap rumah sakit kepada UTD pusat dilakukan dengan via telepon ataupun dapat datang secara langsung ke UTD pusat, namun untuk permohonan permintaan darah dari rumah sakit yang tidak memiliki UTD cabang diharuskan datang langsung ke UTD pusat untuk mengajukan permohonan permintaan darah tersebut. Hal tersebut mengakibatkan data yang masuk antara keduanya dapat berbeda, sedangkan proses permohonan permintaan darah yang dijalankan sama. Dan dapat mengakibatkan perbedaan prioritas antara permintaan via telepon dan dengan datang langsung ke UTD pusat, mengingat apabila datang langsung pemohon yang meminta darah membutuhkan waktu selama menuju ke UTD pusat. Selain itu, terdapat tidak jelasnya jumlah stok darah yang ada di UTD pusat

Kota Malang setiap harinya yang juga dapat menghambat waktu permohonan permintaan darah karena harus dilakukan pengecekan ulang terkait ketersediaan darah tersebut. Selama ini proses manajemen darah yang dilakukan antara pemohon dan UTD pusat Kota Malang dari awal permohonan permintaan darah hingga darah diterima masih belum memiliki informasi akurat dalam proses manajemen darah, proses apa saja yang dilakukan agar darah akhirnya sampai kepada pemohon.

Untuk sisi donor darah, UTD Kota Malang masih menggunakan cara manual yang membutuhkan waktu yang tidak sedikit. Proses donor darah yang dilakukan adalah pendonor harus datang langsung ke UTD pusat untuk melakukan beberapa tahapan donor darah yaitu seperti mengisi data diri dan beberapa kuisioner dari formulir yang telah disediakan, menunggu giliran apabila terdapat beberapa antrian, melakukan pengecekan petugas dari hasil kuisioner yang telah diisi, melakukan pengecekan kesehatan pendonor, dan setelah itu apabila pendonor telah menjalankan semua tahapan yang ditentukan, petugas UTD dapat melakukan tahapan pengambilan darah dari pendonor tersebut. Serta lamanya waktu untuk mendapatkan darah apabila stok darah tidak ada di UTD, karena penyebaran informasi yang menggunakan media sosial dan dengan tingkat keakuratan yang diragukan.

Berdasarkan permasalahan yang telah dijelaskan tersebut, maka solusi yang dapat diberikan adalah dengan membuat sebuah sistem yang dapat manajemen data-data permohonan permintaan darah apabila UTD cabang dan rumah sakit lainnya mengajukan permohonan, menyamaratakan atau melihat keurgensian dari permohonan permintaan diantara keduanya apabila permohonan datang secara bersamaan, memberikan suatu informasi mengenai stok darah secara akurat yang dapat dilihat secara *real time*, adanya transparansi pengiriman darah hingga sampai ke rumah sakit tersebut, meminimalisirkan waktu dalam proses pengajuan donor darah bagi pendonor, dan mendapatkan informasi akurat mengenai permohonan darah yang stok darah tersebut sedang tidak terdapat pada UTD, yang berakibat bagi pemohon memperoleh pendonor dengan lebih cepat. Maka dengan adanya penelitian yang berjudul “Pengembangan Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD) PMI Malang” diharapkan dapat memberikan solusi terbaik dari permasalahan yang ada pada UTD PMI Malang.

1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, maka dapat dirumuskan masalah yang akan diselesaikan dalam penelitian ini yaitu:

1. Bagaimana analisis kebutuhan sistem aplikasi manajemen pada Unit Transfusi Darah (UTD) PMI Malang?
2. Bagaimana hasil perancangan sistem aplikasi manajemen pada Unit Transfusi Darah (UTD) PMI Malang?

3. Bagaimana implementasi sistem aplikasi manajemen pada Unit Transfusi Darah (UTD) PMI Malang?
4. Bagaimana hasil pengujian sistem aplikasi manajemen pada Unit Transfusi Darah (UTD) PMI Malang?

1.3 Tujuan

Berdasarkan rumusan masalah penelitian yang telah dipaparkan, maka tujuan yang dimaksudkan dalam penelitian ini adalah:

1. Mengidentifikasi persyaratan fungsional maupun non fungsional dari sistem aplikasi manajemen pada Unit Transfusi Darah (UTD) PMI Malang.
2. Merancang sistem aplikasi manajemen UTD dengan menggunakan pemodelan berorientasi objek.
3. Mengimplementasikan sistem aplikasi manajemen UTD dengan menggunakan *framework codeigniter*.
4. Menguji sistem aplikasi manajemen UTD sesuai dengan persyaratan fungsional dan non fungsional.

1.4 Manfaat

Manfaat secara umum yang didapat dari sistem aplikasi ini, pihak UTD PMI Kota Malang dapat memanajemen data pendonor yang ada secara akurat dan efisien, serta dapat mengetahui proses manajemen darah dari UTD pusat hingga ke pemohon permintaan darah, dapat mengurangi jumlah penggunaan formulir pengajuan donor darah, karena formulir akan diimplementasikan ke dalam sistem yang dapat diakses dimanapun melalui perangkat yang terhubung ke internet.

Manfaat secara khusus yang didapat diantaranya adalah terdapat fitur yang dapat melakukan permintaan darah secara *real-time* yang berarti apabila beberapa UTD cabang maupun pemohon meminta darah saat waktu yang bersamaan maka dapat dilihat siapa yang terlebih dahulu memohon permintaan dan dilihat melalui keakuratan surat dokter yang dilampirkan saat pembuatan formulir permintaan darah agar darah dapat segera di proses untuk dikirimkan.

1.5 Batasan masalah

Adapun batasan masalah pada penelitian ini agar permasalahan yang dirumuskan lebih terfokus adalah sebagai berikut:

1. Sistem aplikasi manajemen UTD akan dibangun dengan menggunakan pemrograman berbasis web.
2. Observasi dilakukan di UTD PMI Kota Malang sehingga masalah yang terjadi hanya pada lingkup Kota Malang.

3. Data dan parameter yang ada dalam sistem hanya bersumber dari UTD PMI Kota Malang sehingga fitur sistem menyesuaikan kebutuhan UTD PMI.

1.6 Sistematika pembahasan

Sistematika pembahasan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusun tugas akhir secara garis besar yang meliputi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, serta sistematika penulisan yang digunakan dalam penyusunan skripsi ini.

BAB II LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan akan menguraikan kajian pustaka, dasar-dasar teori yang melandasi penulisan dan penelitian.

BAB III METODOLOGI PENELITIAN

Bab metode penelitian membahas mengenai metodologi penelitian pengerjaan skripsi ini. Bab ini terdiri dari beberapa sub bab antara lain tipe penelitian, strategi dan rancangan penelitian, studi kepastakaan, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengambilan keputusan dan saran.

BAB IV ANALISIS KEBUTUHAN

Bab analisis kebutuhan akan membahas gambaran dari kondisi pada proses manajemen yang ada saat ini serta menganalisis kebutuhan sistem untuk dijadikan landasan dalam melakukan perancangan sistem.

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab perancangan membahas tentang hasil rancangan dari sistem yang akan dikembangkan. Perancangan sistsem berdasarkan analisis kebutuhan dan akan dirancang menggunakan perancangan *diagram sequence*, perancangan basis data, perancangan *class diagram*, perancangan algoritme dan perancangan antarmuka. Terdapat beberapa diagram yang digunakan sebagai acuan untuk proses implementasi sistem. Untuk implementasi membahas tentang implementasi sistem berupa bahasa pemrograman yang digunakan serta prosedur-prosedur yang dilakukan pada pembuatan sistem dan disesuaikan dengan rancangan yang telah dirancang.

BAB VI PENGUJIAN

Bab pengujian berisi tentang pengujian dan analisis terhadap sistem yang dibangun. Pengujian dan analisis akan dijelaskan dalam bentuk gambar-gambar dari proses pengujian dan analisis sistem.

BAB VII PENUTUP

Bab penutup berisi kesimpulan yang didasarkan atas pengujian dan analisis yang dilakukan dalam proses penelitian, serta saran-saran dari hasil yang diperoleh yang diharapkan dapat bermanfaat dalam pengembangan selanjutnya.



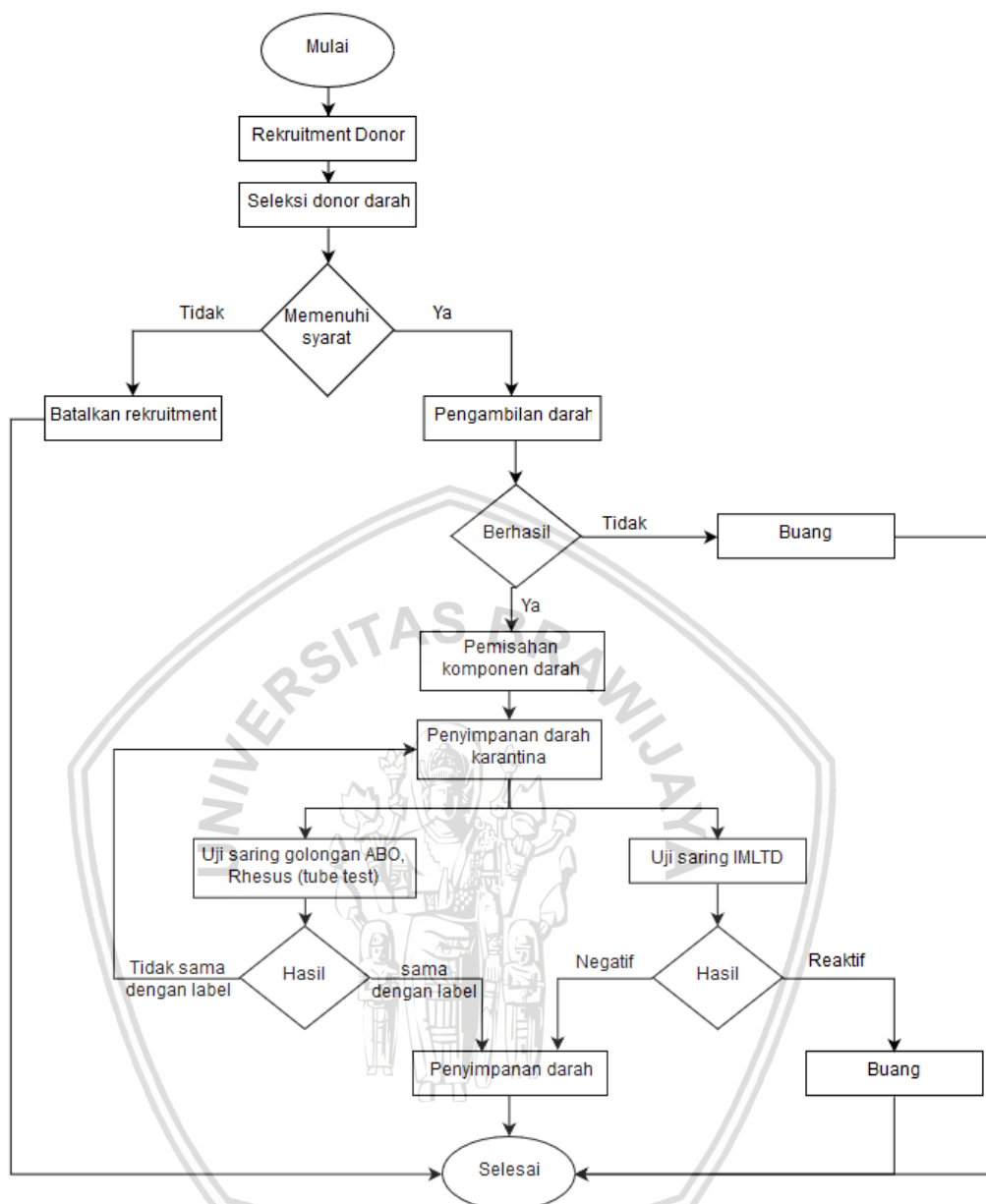
BAB 2 LANDASAN KEPUSTAKAAN

2.1 Manajemen

Manajemen adalah pengendalian dan pemanfaatan daripada semua faktor dan sumber daya yang menurut suatu perencanaan (*planning*), diperlukan untuk mencapai atau menyelesaikan suatu prapta (*objective*) atau tujuan-tujuan tertentu (Atmosudirdjo, 1986). Selain itu, manajemen dapat didefinisikan juga sebagai kemampuan atau ketrampilan untuk memperoleh sesuatu hasil dalam rangka pencapaian tujuan melalui kegiatan-kegiatan orang lain (Siagian, 1989) atau juga dapat dikatakan sebagai pencapaian tujuan yang ditetapkan terlebih dahulu dengan mempergunakan kegiatan orang lain (Manullang, 2005). Dari definisi-definisi tersebut di atas, ada tiga hal penting dalam definisi-definisi tersebut. Pertama, ada tujuan yang hendak dicapai; kedua, tujuan yang hendak dicapai membutuhkan tenaga orang lain; dan ketiga, aktivitas orang lain tersebut harus dibimbing dan diawasi atau dikontrol.

2.1.1 Manajemen Unit Transfusi Darah (UTD)

Manajemen Unit Transfusi Darah (UTD) merupakan suatu proses untuk pengorganisasian, perencanaan, mengendalikan, dan mengarahkan UTD agar dapat mencapai suatu tujuan yang telah ditetapkan. Sesuai dengan pengertian yang telah dijabarkan, maka UTD mempunyai beberapa perencanaan atau pengarahan dalam proses pengorganisasian darah, yaitu pengarahan untuk donor darah bagi pendonor dan permintaan darah bagi petugas rumah sakit maupun UTD setiap rumah sakit.

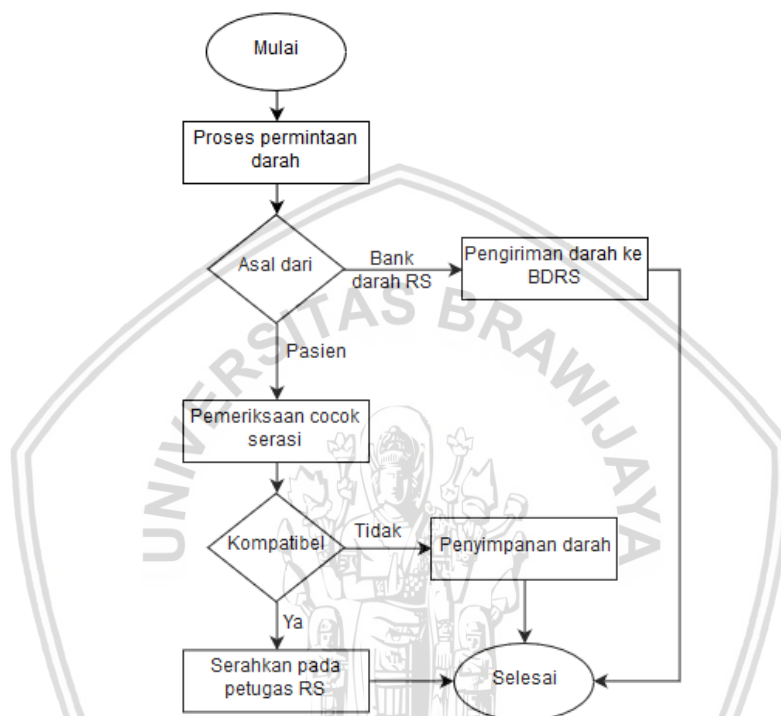


Gambar 2.1 Bagan Alur Donor Darah Unit Transfusi Darah (UTD) PMI Malang

Sumber: UTD PMI Malang

Dalam Gambar 2.1 diatas merupakan alur donor darah dengan tahapan pertama adalah pendonor diharuskan melakukan rekrutmen donor dengan mengisi formulir yang disediakan UTD, selanjutnya akan dilakukan proses seleksi donor darah dengan pengecekan kesehatan pendonor oleh petugas UTD. Jika hasil pengecekan memenuhi syarat maka akan dilakukan proses pengambil darah namun, apabila tidak memenuhi syarat maka rekrutmen akan dibatalkan. Selanjutnya setelah proses pengambilan darah berhasil maka akan dilakukan pemisahan komponen darah oleh petugas laboratorium. Apabila pengambilan darah tidak berhasil akibat beberapa faktor, maka darah akan otomatis dibuang. Darah yang telah melakukan pemisahan komponen maka akan disimpan dalam

karantina dan akan dilakukan uji saring golongan ABO dan rhesus serta uji saring Infeksi Menular Melalui Transfusi Darah (IMLTD). Apabila uji saring golongan ABO dan rhesus sama dengan label golongan darah maka akan melakukan penyimpanan darah, namun apabila uji saring golongan ABO dan rhesus tidak sama dengan label golongan darah maka darah diberikan ke penyimpanan darah karantina. Selanjutnya apabila darah dilakukan uji saring Infeksi Menular Melalui Transfusi Darah (IMLTD) dengan hasil reaktif maka darah akan dibuang, namun apabila hasil negatif maka akan dilakukan penyimpanan darah.



Gambar 2.2 Bagan Alur Permintaan Darah Unit Transfusi Darah (UTD) PMI Malang

Sumber: UTD PMI Malang

Berdasarkan Gambar 2.2 diatas merupakan alur permintaan darah UTD PMI Malang yang mempunyai tahapan pertama adalah petugas rumah sakit meminta darah ke UTD PMI dengan mengisi formulir permintaan dan melampirkan surat dokter yang bertanggung jawab atas permintaan darah. Apabila permintaan berasal dari bank darah rumah sakit maka darah akan dikirimkan langsung ke bank darah rumah sakit tersebut. Namun, apabila permintaan berasal dari pasien maka akan dilakukan pemeriksaan cocok serasi darah terlebih dahulu apabila darah kompatibel/serasi maka darah diserahkan kepada petugas rumah sakit, namun apabila darah tidak kompatibel/serasi maka darah yang tidak serasi tersebut akan disimpan ke dalam penyimpanan darah.

2.2 Software Development Life Cycle (SDLC)

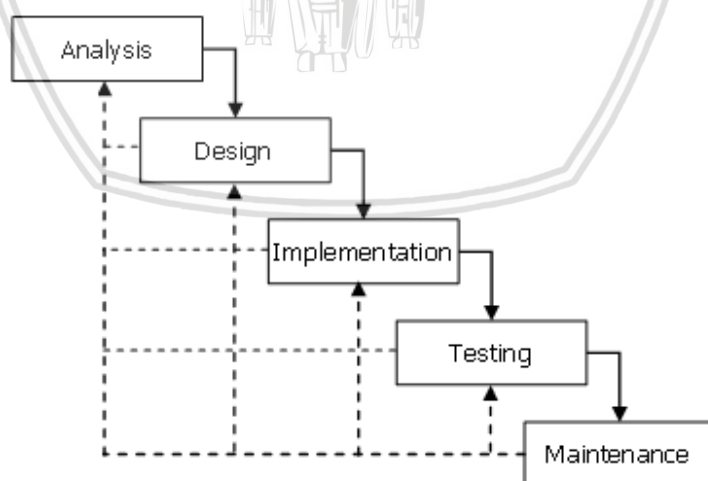
Software Development Life Cycle (SDLC) merupakan serangkaian aktivitas yang dilakukan untuk menciptakan produk perangkat lunak. Ada beragam proses pengembangan perangkat lunak, namun harus memenuhi empat aktivitas yang menjadi dasar perangkat lunak, yaitu: (Sommerville, 2011)

1. *Software specification*, pendefinisian fungsional dan batasan perangkat lunak.
2. *Software design and implementation*, menghasilkan perangkat lunak yang sesuai dengan spesifikasi.
3. *Software validation*, melakukan validasi untuk memastikan bahwa perangkat lunak sesuai dengan kebutuhan aktor.
4. *Software evolution*, mengembangkan *software* dengan memperbaiki sesuai dengan perubahan kebutuhan aktor.

Di dalam SDLC terdapat beberapa model, diantaranya adalah model *waterfall* yang akan digunakan dalam penelitian ini.

2.2.1 Model Waterfall

Model pertama dari proses pengembangan perangkat lunak berasal dari proses rekayasa sistem yang lebih umum. Model ini terlihat pada Gambar 2.2 dengan bentuk seperti air terjun dari satu fase ke fase yang lain, model ini dikenal sebagai model *waterfall* atau siklus hidup perangkat lunak. Model *waterfall* adalah contoh dari sebuah proses perencanaan. Pada prinsipnya, awal proses adalah merencanakan dan menjadwalkan semua proses aktivitas sebelum memulai mengerjakannya.



Gambar 2.3 Model waterfall

Sumber: Bassil (2012)

Berikut tahapan-tahapan utama dalam model *waterfall* yang mencerminkan kegiatan pengembangan yang mendasar:

1. *Analysis*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh *software* yang akan dibangun. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan sebagainya.

2. *Design*

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti mengenai domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *User interface* dan sebagainya. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditujukan kepada *user*. Proses *software design* untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk "*blueprint*" *software* sebelum melakukan *coding*. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

3. *Implementation*

Untuk dapat dimengerti oleh komputer, maka desain harus diubah menjadi bentuk yang dapat dimengerti oleh komputer, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design*.

4. *Testing*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari kesalahan, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

5. *Maintenance*

Tahap ini dilakukan setelah melakukan tahap-tahap sebelumnya yang telah didefinisikan. Dimana proses *maintenance* ini adalah proses pemeliharaan sistem yang dilakukan dengan berkala, untuk mengecek kinerja sistem apakah tetap baik atau tidak.

2.3 Framework Codeigniter

Codeigniter adalah salah satu *framework* PHP yang dapat membantu memudahkan dalam penulisan kode dan membuat struktur kode menjadi lebih baik, *codeigniter* dapat menyederhanakan jumlah kode yang akan ditulis sehingga

kode program dapat lebih mudah untuk dibaca dan diperbaiki, hal ini dapat membantu dalam pembuatan *website* yang memiliki struktur yang koheren (Upton, 2007). *Codeigniter* menggunakan pendekatan *Model View Control* (MVC) yang memisahkan proses pengembangan aplikasi berdasarkan komponen yang terdiri dari komponen yang berfungsi untuk membangun sebuah aplikasi seperti antarmuka, manipulasi data, dan bagian yang menhadi kontrol aplikasi, komponen tersebut membentuk suatu pola MVC dalam suatu aplikasi yaitu *model*, *view*, *controller* (Sidik, 2012). MVC akan memudahkan dalam membuat program karena memiliki kelebihan *reusable* pada fungsi program (Sofwan, 2012). Terdapat tiga jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu: (Blanco dan Upton, 2008)

1. *Model* merepresentasikan data aplikasi yang dibuat, dimasukkan ke dalam database, dalam file XML atau dimanapun. Juga, interaksi dengan *database* dilakukan dalam *model*. Sebagai contoh, model dapat digunakan dalam mengambil, mengubah, menyisipkan, dan menghapus data dari *database*. Semua tindakan yang membutuhkan aplikasi untuk mengakses *database* harus dimasukkan ke dalam *model*.
2. *View* bertanggung jawab untuk menampilkan data untuk para pengunjung situs, atau aktor aplikasi. Tidak terdapat logika pemrograman dan *query insert* atau *update* yang dijalankan, meskipun akses data terjadi pada file-file ini. *View* hanya berfungsi untuk menunjukkan hasil dari dua lainnya (Controller dan Model).
3. *Controller* bertindak dalam menjembatani antara *model* dengan *view*. Terdapat logika pemrograman dalam *controller*.

2.4 Unified Modeling Language (UML)

UML merupakan singkatan dari *Unified Modeling Language* yang jika diterjemahkan ke Bahasa Indonesia menjadi bahasa pemodelan standar. Terjemahan dari singkatan tersebut menunjukkan bahwa dalam pemodelan menggunakan bahasa yang terstandar. UML merupakan sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek (Whitten, 2004). UML memiliki beberapa komponen yang biasa digunakan dalam merancang suatu perangkat lunak.

Semakin kompleks bentukan sistem yang akan dibuat, maka semakin sulit komunikasi antara orang-orang yang saling terkait dalam pembuatan dan pengembangan perangkat lunak yang akan dibuat. Pada masa lalu, UML mempunyai peranan sebagai *software blueprint* (gambaran) language untuk analisis sistem, *designer*, dan *programmer*. Sedangkan pada saat ini, merupakan bagian dari *software trade* (bisnis *software*). UML memberikan jalur komunikasi dari sistem analisis kemudian *designer*, lalu *programmer* mengenai rancangan *software* yang akan dikerjakan.

2.4.1 Use Case Diagram

Use Case merupakan unit fungsi yang dihasilkan oleh suatu sistem. Tujuan dari *use case diagram* adalah untuk membantu proses *development team* dalam memvisualisasikan kebutuhan fungsional sistem, termasuk hubungan antar aktor untuk melakukan suatu proses serta hubungan antara *use case* yang berbeda (Bell, 2003). Dengan adanya *use case* maka dapat diketahui gambaran tentang keterlibatan satu atau lebih aktor dengan suatu sistem. *Use case diagram* secara umum berisi bagian-bagian sebagai berikut: (Booch, 2005)

1. Subjek, merupakan sistem atau subsistem yang berisi *use cases* dan aktor.
2. *Use cases*, merupakan tindakan-tindakan yang dilakukan oleh aktor. *Use case* ditulis dengan menggunakan kata kerja di awal kalimat.
3. Aktor, merupakan entitas yang berinteraksi dengan *use case*. Aktor dijadikan sebagai peranan yang dapat berinteraksi dengan sistem. Aktor dapat berupa manusia, sebuah perangkat keras atau sebuah sistem yang berinteraksi dengan sistem.

2.4.2 Sequence Diagram

Sequence diagram merupakan gambaran hubungan antara objek satu dengan objek lainnya. *Sequence diagram* pada umumnya digunakan untuk menggambarkan suatu skenario atau langkah-langkah yang dilakukan oleh aktor ataupun sistem yang merupakan sebuah respon dari *use case* tertentu (Booch, 2005). Untuk membuat *sequence diagram* maka harus mengetahui obyek-obyek yang terlibat dalam sebuah *use case* beserta metode yang dimiliki oleh kelas yang diinstansiasi oleh obyek tersebut. Perancangan *sequence diagram* minimal sebanyak jumlah *use case* yang telah didefinisikan interaksi alur pesannya dan *use case* yang memiliki proses sendiri.

2.4.3 Class Diagram

Class diagram merepresentasikan struktur dari sistem dengan cara mendeskripsikan beberapa *class* yang akan dibangun pada sistem. Masing-masing *class* memiliki atribut, operasi atau *method* (Rosa dan Shalahuddin, 2011). *Class diagram* secara umum berisi beberapa bagian, antara lain: (Booch, 2005)

1. *Classes*, merupakan deskripsi dari himpunan objek-objek yang memiliki kesamaan atribut, operasi, dan relasi. *Class* digambarkan dalam bentuk empat persegi panjang.
2. *Interfaces*, merupakan kumpulan operasi yang digunakan untuk spesifikasi sebuah *class* atau komponen. *Interfaces* digambarkan dalam bentuk *stereotype class*.
3. Relasi, merupakan sebuah koneksi antara dua hal. Dalam pemodelan *object-oriented*, relasi memiliki tiga macam relasi, yaitu dependensi, generalisasi, dan asosiasi.

2.5 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan sebuah *set* dari langkah-langkah dimana kita dapat menempatkan desain spesifik *test case* dan metode pengujian. Metode-metode ini bertujuan untuk menyediakan mekanisme yang dapat memastikan kelengkapan dari pengujian dan kemungkinan tertinggi untuk menemukan kesalahan yang ada pada perangkat lunak yang telah dibuat (Pressman, 2001). Semuanya menyediakan sebuah aturan untuk pengujian dan juga memiliki karakteristik dasar sebagai berikut: (Sommerville, 2011)

1. Untuk melakukan pengujian yang efektif, maka harus dilakukan *review* teknis yang efektif. Dengan melakukan ini, banyak kesalahan akan dieliminasi sebelum pengujian dimulai.
2. Pengujian dimulai pada level komponen dari seluruh sistem berbasis komputer.
3. Teknik pengujian yang berbeda sesuai untuk pendekatan rekayasa perangkat lunak yang berbeda dan pada poin-poin yang berbeda pada satu waktu.
4. Pengujian dilakukan oleh pengembang perangkat lunak dan sebuah grup pengujian yang independen (untuk proyek besar).
5. Pengujian dan penelusuran merupakan aktivitas yang berbeda, tetapi tetap harus dimasukkan ke dalam strategi pengujian.

2.5.2 Whitebox Testing

White-box testing adalah pendekatan dalam pengujian program dimana tes didasarkan pada pengetahuan tentang struktur program dan komponennya (Pressman, 2010). *Whitebox Testing* membutuhkan waktu pengujian yang lebih lama karena membutuhkan ketelitian dari penguji serta kemampuan teknis pemrograman bagi para pengujinya (Rizky, 2011). *Whitebox testing* mempunyai berbagai jenis pengujian. Namun, pada penelitian ini jenis yang digunakan adalah pengujian *basis path testing*. Pengujian *basis path testing* adalah jenis pengujian yang dilakukan dengan menggunakan *cyclomatic complexity* dengan tujuan untuk mengetahui aliran control logika yang digambarkan dengan menggunakan *flow graph* yang dapat menghasilkan jalur independen (Pressman, 2010).

2.5.3 Blackbox Testing

Blackbox Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program Shihab (2011). *Blackbox Testing* berupaya untuk menemukan kesalahan dalam kategori berikut: fungsi yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan perilaku atau kinerja dan kesalahan inisialisasi dan penghentian (Pressman, 2010).

Jenis *blackbox testing* yang digunakan pada penelitian ini adalah scenario-based testing dan equivalence partitioning. Scenario-based testing merupakan

pengujian yang mengacu kepada aktifitas *user* terhadap sistem, tetapi tidak untuk aktifitas sistem itu sendiri (Pressman, 2010). *Equivalence Partitioning* didasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak dalam beberapa kelas data (Pressman, 2010). Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi boolean.



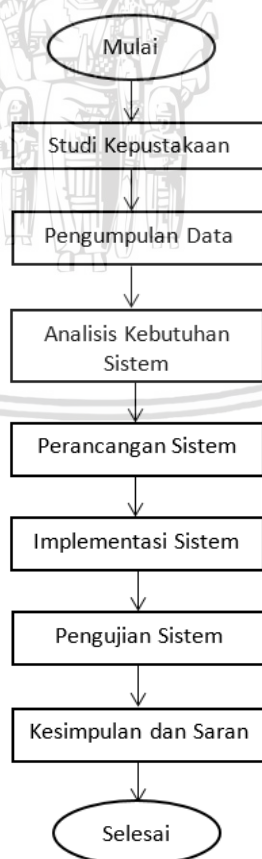
BAB 3 METODOLOGI PENELITIAN

3.1 Tipe Penelitian

Tipe penelitian yang akan digunakan dalam penelitian ini bertipe Implementatif – Pengembangan. Penelitian Implementatif – Pengembangan yang dimaksudkan adalah penelitian yang pada hasil akhirnya akan menghasilkan sebuah produk perangkat lunak dengan menerapkan prinsip-prinsip rekayasa kebutuhan yang didefinisikan. Produk perangkat lunak tersebut adalah sistem aplikasi manajemen Unit Transfusi Darah (UTD) Malang.

3.2 Strategi dan Rancangan Penelitian

Bab ini menjelaskan langkah-langkah yang dilakukan untuk membuat sistem aplikasi manajemen UTD PMI. Pengambilan dasar teori menggunakan metode landasan kepustakaan dan wawancara kepada pihak UTD PMI. Data tersebut diperoleh melalui studi kepustakaan. Penjelasan mengenai langkah-langkah yang akan dilakukan selama pengerjaan skripsi ini akan di jelaskan pada bab ini. Langkah-langkah yang akan dilakukan dimulai dari studi kepustakaan, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, dan kesimpulan serta saran. Seluruh langkah-langkah tersebut diilustrasikan pada Gambar 3.1 *diagram* alir dibawah ini.



Gambar 3.1 Diagram Alir Metode Penelitian

3.2.1 Studi Kepustakaan

Studi kepustakaan adalah metode untuk mendalami konsep yang digunakan dalam membangun sebuah sistem. Studi kepustakaan berisi landasan teori yang terkait dengan konsep tersebut. Hal ini sangat perlu dilakukan agar pengetahuan dasar untuk membangun sebuah sistem terpenuhi dengan baik. Studi kepustakaan yang digunakan dalam membangun sistem aplikasi manajemen ini didapat dari buku, skripsi, artikel ilmiah, dan informasi yang tersedia di internet baik berupa proyek maupun artikel. Studi kepustakaan yang digunakan meliputi di bawah ini :

1. Manajemen Unit Transfusi Darah
2. *Software Development Life Cycle* (SDLC)
 - a. Model *Waterfall*
3. *Unified Modeling Language* (UML)
4. *Framework Codeigniter*
5. Pengujian Perangkat Lunak

3.2.2 Pengumpulan Data

Pengumpulan data adalah proses dalam mendapatkan suatu data terkait yang dibutuhkan pada saat penelitian. Data ini didapat melalui pengamatan terhadap dokumen laporan stok darah, alur distribusi darah, manajemen darah, alur donor darah, persyaratan permintaan darah, dan persyaratan donor darah.

3.2.3 Analisis Kebutuhan

Analisis kebutuhan adalah metode untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (requirement) perangkat lunak dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan yang ada sehingga dapat diketahui implementasi perangkat lunak yang akan digunakan. Kebutuhan-kebutuhan yang ditentukan akan dikategorikan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan-kebutuhan yang ditentukan berasal dari wawancara langsung dengan pihak yang terlibat. Kebutuhan yang didapat akan dibuat pemodelan dengan menggunakan salah satu diagram UML yaitu *use case diagram*. Pemodelan kebutuhan dilakukan untuk menjelaskan kebutuhan sistem yang akan dibangun dan siapa saja yang berinteraksi dengan sistem tersebut. Setelah itu dilakukan pembuatan *use case scenario* untuk menjelaskan cara kerja sistem berdasarkan *use case diagram* yang telah dibuat. Dari proses ini akan menghasilkan suatu gambaran sistem yang dapat memberikan kemudahan dalam memahami sistem yang dibuat serta proses-proses yang ada. Proses analisis kebutuhan sistem aplikasi manajemen Unit Transfusi Darah (UTD) PMI Malang meliputi:

1. **Elisitasi Kebutuhan**
Pada tahap ini elisitasi kebutuhan dilakukan proses wawancara untuk mengetahui apa saja kebutuhan sistem yang akan dibangun. Wawancara akan dilakukan kepada staff UTD PMI Kota Malang, Bapak Agus Pras, bagian penelitian dan pengembangan.
2. **Deskripsi Sistem**
Tahap deskripsi sistem berisi mengenai penjelasan sistem yang akan dirancang dan diimplementasikan. Sistem akan seperti apa dan apa saja yang terdapat dalam sistem serta apa saja yang dapat dilakukan oleh sistem dan kaitan dengan topik yang ada.
3. **Identifikasi Aktor**
Tahap identifikasi aktor bertujuan untuk memperoleh daftar aktor yang akan berinteraksi langsung dengan sistem. Pada tahap ini akan mendefinisikan siapa saja aktor yang terlibat di dalam sistem dan apa saja peran dari masing-masing aktor tersebut.
4. **Spesifikasi kebutuhan**
Pada proses ini kebutuhan yang sudah didapatkan akan didefinisikan hingga menjadi kebutuhan yang spesifik. Dalam proses ini juga melakukan identifikasi aktor yang akan menggunakan sistem ini dan juga kebutuhan yang sudah didapatkan akan dijadikan kedalam dua jenis kebutuhan yaitu fungsional dan non fungsional dengan tujuan untuk mengetahui secara rinci aktifitas apa saja yang dapat dilakukan oleh aktor maupun sistem dan apa saja batasan-batasan sistem yang akan dibangun.
5. **Pemodelan Kebutuhan Sistem**
Pemodelan kebutuhan merupakan tahapan dimana hasil dari analisis kebutuhan sistem digambarkan kedalam bentuk use case diagram. Hal ini bertujuan agar gambaran dari hubungan aktor dan fungsional sistem menjadi lebih jelas. Selanjutnya hasil dari fungsional sistem tersebut dijabarkan kedalam use case scenario dengan tujuan agar fungsional lebih jelas dalam pelaksanaannya

3.2.4 Perancangan Sistem

Hasil dari analisis kebutuhan sistem sebelumnya, selanjutnya dirancang menjadi suatu model atau rancangan perangkat lunak dengan menggunakan teknik atau prinsip *Object Oriented Programming* sehingga hasil dari perancangan perangkat lunak dapat diwujudkan menjadi suatu perangkat lunak yang sebenarnya. Perancangan yang akan dilakukan sebagai berikut:

1. **Perancangan arsitektur**
Perancangan yang akan dibangun dijelaskan kepada bentuk UML (*Unified Modeling Language*) berupa *sequence diagram* dan *class diagram*. *Sequence diagram* yang akan diambil sebanyak 3 sampel penting dari perangkat lunak. Untuk *Class diagram* akan menjelaskan pengelompokan class yang ada di perancangan.

2. Perancangan struktur

Selanjutnya pada perancangan struktur, akan dibuat perancangan yang menjelaskan mengenai basis data yaitu *Physical Data Model* (PDM) yang menjadi dasar dalam pembangunan basis data.

3. Perancangan algoritme

Merupakan perancangan yang menjelaskan tentang 3 penjelasan dari algoritme dari sistem.

4. Perancangan antar muka

Dalam perancangan antar muka akan dibuat beberapa tampilan dari perangkat lunak. Hal ini bertujuan untuk memberikan gambaran mendasar dari antarmuka yang akan diimplementasikan kedalam sistem.

3.2.5 Implementasi Sistem

Implementasi akan dibuat dengan mengacu pada tahap perancangan. Implementasi sistem dibuat dengan menggunakan bahasa pemrograman web yang terdiri dari: HTML, CSS, PHP, dan Javascript. *Framework* yang digunakan adalah berupa *Codeigniter*. Model pengembangan sistem aplikasi manajemen berbasis web ini menggunakan *waterfall* dan menggunakan MySQL sebagai manajemen basis data.

3.2.6 Pengujian Sistem

Pengujian merupakan bagian dari siklus pengembangan suatu perangkat lunak. Tujuan dari pengujian ini adalah untuk memberikan jaminan kepada aktor bahwa aplikasi yang dibangun telah sesuai dengan kebutuhan atau spesifikasi yang diinginkan aktor. Pengujian sistem yang akan dilakukan terdiri dari 3 pengujian, yaitu pengujian *black box*, pengujian *white box*, dan pengujian *compatibility*.

Pengujian fungsional perangkat lunak akan dilakukan menggunakan *black box testing*. *Black box testing* dilakukan menggunakan pengujian validasi yang akan menguji semua yang berkaitan dengan fitur-fitur pada sistem.

Pengujian *white box* pada penelitian ini difokuskan kepada pengujian unit yang akan digunakan dalam menguji suatu komponen tertentu dengan memastikan bahwa algoritme pada suatu komponen bekerja dengan baik. Teknik pengujian yang digunakan dengan menggunakan *basis path testing*. Teknik pengujian ini dibuat dengan menggunakan rancangan algoritme atau kode program sebagai dasar dalam membentuk suatu *flow graph*. Setelah itu, berdasarkan hasilnya akan ditentukan *cyclomatic complexity* serta jalur-jalur independen yang terdapat pada fungsi terkait.

3.2.7 Kesimpulan dan Saran

Pengambilan kesimpulan diberikan setelah semua tahapan telah dilakukan. Kesimpulan yang diambil berdasarkan pada hasil observasi awal, pengumpulan data, analisis kebutuhan, perancangan, implementasi, pengujian dan hingga analisis hasil penelitian. Kesimpulan penelitian akan menjawab permasalahan penelitian. Hasil kesimpulan diberikan berdasarkan keseluruhan proses

pembuatan sistem aplikasi manajemen UTD PMI untuk menunjukkan apakah masalah penelitian terjawab dan tujuan penelitian tercapai. Dengan adanya kesimpulan dapat memberi saran atau pertimbangan untuk perancangan perangkat lunak selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

Bab analisis kebutuhan menjelaskan mengenai kebutuhan sistem aplikasi manajemen unit transfusi darah pada PMI Malang. Analisis kebutuhan merupakan salah satu tahap yang harus dilakukan sebelum melakukan tahap perancangan dan implementasi perangkat lunak. Tahap analisis kebutuhan pada penelitian ini diawali dengan elisitasi kebutuhan, analisis kebutuhan.

4.1 Elisitasi Kebutuhan

Elisitasi kebutuhan dilakukan untuk mengetahui permasalahan yang terjadi pada UTD PMI Kota Malang. Tahapan ini dilakukan dengan melakukan wawancara kepada staff bidang mutu dan pelayanan UTD dengan tujuan untuk mengetahui permasalahan secara lebih mendalam.

Berdasarkan hasil wawancara yang telah dilakukan, diketahui terdapat masalah yang ada. Masalah-masalah tersebut adalah permohonan permintaan darah dari rumah sakit dengan berbeda cara antara UTD cabang dan yang tidak memiliki UTD cabang. UTD cabang setiap rumah sakit dapat melakukan permohonan melalui via telepon ataupun datang secara langsung ke UTD pusat, namun untuk rumah sakit yang tidak memiliki UTD cabang diharuskan datang langsung ke UTD pusat. Terdapat tidak jelasnya jumlah stok darah yang ada di UTD pusat Kota Malang setiap harinya yang juga dapat menghambat waktu permohonan permintaan darah, dan tidak adanya transparansi proses manajemen darah dari awal permohonan permintaan darah hingga darah diterima.

Donor darah dilakukan dengan cara manual yang mengakibatkan pemanfaatan waktu yang tidak baik, lamanya waktu untuk mendapatkan darah apabila stok darah tidak ada di UTD dengan penyebaran informasi yang menggunakan media sosial yang mempunyai tingkat keakuratan yang masih diragukan.

Dari permasalahan tersebut, maka solusi yang dapat diberikan adalah sistem dapat memanajemen data-data permohonan permintaan darah tiap rumah sakit, menyamaratakan atau melihat keurgensian dari permohonan permintaan diantara keduanya apabila permohonan datang secara bersamaan, memberikan informasi mengenai stok darah secara *real time*, adanya transparansi pengiriman darah hingga sampai ke rumah sakit, dan mendapatkan informasi akurat mengenai permohonan darah yang stok darah tersebut sedang tidak terdapat pada UTD.

4.2 Deskripsi Sistem

Sistem aplikasi manajemen unit transfusi darah adalah perangkat lunak yang digunakan dalam melakukan manajemen darah pada unit transfusi darah PMI Malang. Sistem ini harus dapat digunakan dimanapun dan kapanpun karena kebutuhan data darah yang diperlukan masyarakat maupun petugas bersifat *real time* untuk memenuhi permintaan darah serta permohonan pendonoran darah. Sistem juga harus dapat digunakan di berbagai macam perangkat yang memiliki peramban web dan dapat terhubung ke internet. Dalam web ini memiliki fitur

utama yaitu manajemen darah dari permintaan darah hingga darah diterima oleh pasien yang membutuhkan.

Pada sistem terdapat empat aktor. Pada masing-masing aktor mempunyai wewenang hak akses yang berbeda. Bagian *guest* hanya dapat melihat informasi-informasi yang disediakan oleh sistem seperti informasi stok darah. Bagian pendonor dapat mengetahui informasi darah yang dibutuhkan, serta dapat mengajukan permohonan donor darah baik darah normal maupun darah *rhesus* negatif, berbeda dengan petugas rumah sakit yaitu dengan dapat mengajukan permintaan darah serta melihat alur manajemen darah dari permohonan permintaan hingga darah sampai ke tujuan, selanjutnya adalah admin yang bertindak memanajemen seluruh data yang telah dimasukkan.

4.3 Identifikasi Aktor

Identifikasi aktor adalah seseorang atau sistem lain yang terlibat dengan sistem secara langsung. Pada tahap ini mengacu pada tahap analisis kebutuhan secara umum yang telah dilakukan sebelumnya, pada Tabel 4.1 berikut menjelaskan faktor yang terkait dengan sistem.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1.	<i>Guest</i>	<i>Guest</i> merupakan aktor yang dapat melihat halaman awal sistem (halaman <i>login</i>), melihat seluruh informasi yang terdapat dalam sistem, dan dapat melakukan <i>login</i> untuk proses autentifikasi terhadap aktor tersebut ketika ingin menggunakan sistem.
2.	<i>Administrator</i>	<i>Administrator</i> merupakan aktor yang dapat menggunakan sistem dan memiliki otorisasi untuk dapat mengakses sebagian besar fitur aplikasi dan dapat memanajemen seluruh data yang akan dimasukkan atau telah dimasukkan.
3.	Pendonor	Pendonor merupakan aktor yang dapat menggunakan sistem dan memiliki otorisasi untuk dapat melihat stok darah dan dapat mengajukan permohonan donor darah baik darah normal maupun darah rhesus negatif.
4.	Petugas Rumah Sakit	Petugas Rumah Sakit merupakan aktor yang dapat menggunakan sistem dan memiliki otorisasi untuk dapat melihat stok darah dan dapat mengajukan

No	Aktor	Deskripsi
		permintaan darah serta dapat melihat alur manajemen darah dari permohonan permintaan hingga darah sampai ke tujuan.

4.4 Spesifikasi Kebutuhan

Spesifikasi kebutuhan merupakan tahap yang dilakukan untuk mengetahui masalah yang perlu dipecahkan dan mengenali pemangku kepentingan dari sistem. Langkah-langkah yang dilakukan yaitu dengan mempelajari alur aktivitas transfusi darah, disesuaikan dengan gambaran umum sistem, dan dokumen yang telah tersedia.

Terdapat 2 jenis kebutuhan yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional menggambarkan apa saja yang dapat dilakukan aktor dari sistem yang akan dikembangkan, sedangkan kebutuhan non-fungsional menggambarkan kebutuhan yang secara tidak langsung dibutuhkan oleh aktor. Kebutuhan fungsional akan diberi kode SAM-F-XX, spesifikasi kebutuhan fungsional akan diberi kode SAM-F-XX-Y dan kebutuhan non-fungsional diberi kode SAM-NF-XX. SAM merupakan singkatan dari Sistem Aplikasi Manajemen, F merupakan singkatan dari Fungsional, NF merupakan singkatan dari Non-fungsional, XX menunjukkan nomor dari definisi kebutuhan, dan Y menunjukkan nomor dari spesifikasi kebutuhan. Pada Tabel 4.2 dapat dilihat hasil definisi dan spesifikasi kebutuhan fungsional sistem dan Tabel 4.3 hasil kebutuhan non-fungsional sistem.

Tabel 4.2 Kebutuhan Fungsional

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
1	SAM-F-01	Sistem harus mampu menyediakan sarana berupa formulir untuk aktor agar dapat mendaftar ke dalam sistem.	Pendaftaran	Guest
1.1	SAM-F-01-1	Data formulir yang disediakan adalah kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> yang akan digunakan.		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
2	SAM-F-02	Sistem harus mampu menyediakan sarana berupa formulir untuk mengizinkan aktor mengakses sistem sesuai dengan otoritasnya.	Login	Guest
2.1	SAM-F-02-1	Data untuk autentifikasi yang diperlukan adalah <i>username</i> dan <i>password</i> .		
3	SAM-F-03	Sistem harus mampu menyediakan sarana agar aktor yang sudah masuk dapat keluar dari sistem.	Logout	Administrator, Petugas Rumah Sakit, Pendonor
4	SAM-F-04	Sistem harus mampu menampilkan tabel data stok darah secara <i>realtime</i> .	Melihat Stok Darah	Guest
4.1	SAM-F-04-1	Data untuk tabel stok darah akan berisi nama produk, keterangan produk, dan jumlah darah pada semua golongan darah.		
5	SAM-F-05	Sistem harus mampu menampilkan informasi alur donor darah untuk mengetahui alur yang harus dilakukan sebelum melakukan donor darah.	Melihat Alur Donor	Guest
5.1	SAM-F-05-1	Informasi yang dapat dilihat adalah dalam bentuk poin-poin yang menjelaskan proses donor.		
6	SAM-F-06	Sistem harus mampu menampilkan informasi setelah melakukan donor darah.	Melihat Informasi Setelah Donor	Guest
6.1	SAM-F-06-1	Informasi yang dapat dilihat adalah dalam bentuk poin-poin yang menjelaskan apa saja yang		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		harus dilakukan setelah melakukan donor darah.		
7	SAM-F-07	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan pesan dari aktor.	Menambah Pesan Hubungi	Guest
7.1	SAM-F-07-1	Data formulir yang harus diisi adalah nama lengkap, email, nomor telepon dan pesan yang akan dikirimkan.		
8	SAM-F-08	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar pengguna (<i>administrator</i> , petugas rumah sakit, dan pendonor).	Melihat Daftar Pengguna	<i>Administrator</i>
8.1	SAM-F-08-1	Data tabel yang disediakan adalah <i>username</i> dan level pengguna.		
9	SAM-F-09	Sistem harus mampu menyediakan formulir untuk menambahkan stok darah	Menambah stok darah	<i>Administrator</i>
9.1	SAM-F-09-1	Data formulir yang harus disediakan adalah produk, golongan darah, dan jumlah stok darah.		
10	SAM-F-10	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan pengguna (<i>administrator</i> , petugas rumah sakit, dan pendonor).	Menambah Pengguna	<i>Administrator</i>
10.1	SAM-F-10-1	Data formulir yang harus disediakan adalah		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		username, password, dan level pengguna.		
11	SAM-F-11	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada data pengguna (<i>administrator</i> , petugas rumah sakit, dan pendonor).	Mengubah Data Pengguna	<i>Administrator</i>
11.1	SAM-F-11-1	Data yang dapat diubah adalah username, password, dan level pengguna		
12	SAM-F-12	Sistem harus mampu menyediakan sarana untuk menghapus pengguna (<i>administrator</i> , petugas rumah sakit, dan pendonor).	Menghapus Pengguna	<i>Administrator</i>
13	SAM-F-13	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar permintaan darah.	Melihat Daftar Permintaan Darah	<i>Administrator</i>
13.1	SAM-F-13-1	Data tabel yang disediakan adalah tanggal, status, nama rumah sakit, nomor permintaan, golongan darah, jumlah komponen, dan jenis komponen.		
14	SAM-F-14	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan permintaan darah.	Menambah Permintaan Darah	<i>Administrator</i>
14.1	SAM-F-14-1	Data formulir yang harus disediakan adalah nomor permintaan, nama rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter.		
15	SAM-F-15	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan alur pengiriman darah.	Menambah Alur Pengiriman Darah	Administrator
15.1	SAM-F-15-1	Data formulir yang harus disediakan adalah nama rumah sakit, status, dan petugas.		
16	SAM-F-16	Sistem harus mampu menampilkan informasi detail permintaan darah.	Melihat Detail Permintaan Darah	Administrator
16.1	SAM-F-16-1	Informasi yang dapat dilihat untuk detail permintaan darah adalah waktu permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter. Untuk proses darah adalah tanggal, nama rumah sakit, status, dan petugas.		
17	SAM-F-17	Sistem harus mampu menyediakan sarana untuk melakukan perubahan data pada permintaan darah.	Mengubah Data Permintaan Darah	Administrator
17.1	SAM-F-17-1	Data yang dapat diubah adalah nomor permintaan, nama rumah sakit, nama		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, dan jenis kelamin.		
18	SAM-F-18	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar data petugas rumah sakit.	Melihat Daftar Data Petugas Rumah Sakit	Administrator
18.1	SAM-F-18-1	Data tabel yang disediakan adalah nama rumah sakit, nama petugas, alamat rumah sakit, dan telepon rumah sakit.		
19	SAM-F-19	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada data petugas rumah sakit.	Mengubah Data Petugas Rumah Sakit	Administrator
19.1	SAM-F-19-1	Data yang dapat diubah adalah nama rumah sakit, nama penanggung jawab, alamat rumah sakit, dan telepon rumah sakit.		
20	SAM-F-20	Sistem harus mampu menyediakan sarana untuk menghapus data petugas rumah sakit.	Menghapus Data Petugas Rumah Sakit	Administrator
21	SAM-F-21	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar pengajuan donor.	Melihat Daftar Pengajuan Donor	Administrator
21.1	SAM-F-21-1	Data tabel yang disediakan adalah tanggal pengajuan, nomor pengajuan, kode pendonor, nama pendonor, dan golongan darah.		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
22	SAM-F-22	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan pengajuan donor.	Menambah Pengajuan Donor	Administrator
22.1	SAM-F-22-1	Data formulir yang harus disediakan adalah tanggal pengajuan, nomor pengajuan, kode pendonor.		
23	SAM-F-23	Sistem harus mampu menampilkan informasi detail pengajuan donor.	Melihat Detail Pengajuan Donor	Administrator
23.1	SAM-F-23-1	Informasi yang dapat dilihat adalah tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur pendonor, golongan darah, alamat pendonor, telepon pendonor.		
24	SAM-F-24	Sistem harus mampu menyediakan sarana untuk menghapus pengajuan donor.	Menghapus Pengajuan Donor	Administrator
25	SAM-F-25	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar data pendonor.	Melihat Daftar Data Pendonor	Administrator
25.1	SAM-F-25-1	Data tabel yang disediakan adalah kode pendonor, nomor identitas, nama pendonor, umur, dan telepon.		
26	SAM-F-26	Sistem harus mampu menampilkan informasi detail data pendonor.	Melihat Detail Data Pendonor	Administrator

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
26.1	SAM-F-26-1	Informasi yang dapat dilihat adalah kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon dan email.		
27	SAM-F-27	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada data pendonor.		
27.1	SAM-F-27-1	Data yang dapat diubah adalah kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.	Mengubah Data Pendonor	Administrator
28	SAM-F-28	Sistem harus mampu menyediakan sarana untuk menghapus data pendonor.	Menghapus Data Pendonor	Administrator
29	SAM-F-29	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar riwayat donor pendonor.		
29.1	SAM-F-29-1	Data tabel yang disediakan adalah tanggal donor, kode pendonor, status kesehatan, riwayat donor, dan status riwayat penyakit.	Melihat Daftar Riwayat Donor Pendonor	Administrator
30	SAM-F-30	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan riwayat donor.		
30.1	SAM-F-30-1	Data formulir yang harus disediakan adalah kode pendonor, status	Menambah Riwayat Donor	Administrator

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan.		
31	SAM-F-31	Sistem harus mampu menampilkan informasi detail riwayat donor pendonor.	Melihat Detail Riwayat Donor Pendonor	Administrator
31.1	SAM-F-31-1	Informasi yang dapat dilihat adalah waktu donor, kode pendonor, status kesehatan, status donor, status riwayat penyakit, jumlah komponen, dan kesimpulan.		
32	SAM-F-32	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada data riwayat donor.	Mengubah Data Riwayat Donor	Administrator
32.1	SAM-F-32-1	Data yang dapat diubah adalah kode pendonor, status kesehatan, status donor, status riwayat penyakit, jumlah komponen, dan kesimpulan.		
33	SAM-F-33	Sistem harus mampu menyediakan sarana untuk menghapus riwayat donor.	Menghapus Riwayat Donor	Administrator
34	SAM-F-34	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar data butuh darah.	Melihat Daftar Data Butuh Darah	Administrator
34.1	SAM-F-34-1	Data tabel yang disediakan adalah tanggal, nama wali, nomor telepon, golongan darah, faktor rhesus(rh),		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		jumlah komponen, rumah sakit.		
35	SAM-F-35	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan data butuh darah.	Menambah Data Butuh Darah	Administrator
35.1	SAM-F-35-1	Data formulir yang harus disediakan adalah nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.		
36	SAM-F-36	Sistem harus mampu menampilkan informasi detail data butuh darah.	Melihat Detail Data Butuh Darah	Administrator
36.1	SAM-F-36-1	Informasi yang dapat dilihat adalah tanggal permintaan, nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.		
37	SAM-F-37	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada data butuh darah.	Mengubah Data Butuh Darah	Administrator
37.1	SAM-F-37-1	Data yang dapat diubah adalah nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.		
38	SAM-F-38	Sistem harus mampu menyediakan sarana untuk menghapus data butuh darah.	Menghapus Data Butuh Darah	Administrator

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
39	SAM-F-39	Sistem harus mampu menampilkan informasi pemberitahuan baru.	Melihat Pemberitahuan	Administrator
39.1	SAM-F-39-1	Informasi yang dapat dilihat adalah daftar pengajuan permintaan darah baru.		
40	SAM-F-40	Sistem harus mampu menampilkan informasi syarat donor darah untuk mengetahui syarat yang harus dimiliki sebelum melakukan donor.	Melihat Syarat Donor	Pendonor
40.1	SAM-F-40-1	Informasi yang dapat dilihat adalah dalam bentuk poin-poin yang menjelaskan syarat-syarat donor.		
41	SAM-F-41	Sistem harus mampu menyediakan sarana untuk menambahkan pengajuan donor.	Menambah Pengajuan Donor	Pendonor
41.1	SAM-F-41-1	Data yang harus diisi adalah tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, telepon pendonor, tempat lahir, tanggal lahir, golongan darah, umur pendonor, alamat pendonor, dan email.		
42	SAM-F-42	Sistem harus mampu menyediakan sarana untuk mencetak pengajuan donor.	Mencetak Pengajuan Donor	Pendonor
43	SAM-F-43	Sistem harus mampu menampilkan informasi butuh darah sesuai golongan darah yang dapat	Melihat Informasi Butuh Darah	Pendonor

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		didonorkan oleh pendonor.		
43.1	SAM-F-43-1	Informasi yang dapat dilihat adalah tanggal dan waktu dibutuhkan, nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, dan keterangan yang dibutuhkan.		
44	SAM-F-44	Sistem harus mampu menampilkan informasi detail profil pendonor.		
44.1	SAM-F-44-1	Infomasi yang dapat dilihat adalah kode pendonor, <i>username</i> , nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.	Melihat Profil Pendonor	Pendonor
45	SAM-F-45	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada profil pendonor.		
45.1	SAM-F-45-1	Data yang dapat diubah adalah <i>username</i> , <i>password</i> , nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.	Mengubah Profil Pendonor	Pendonor
46	SAM-F-46	Sistem harus mampu menampilkan informasi penghargaan donor.		
46.1	SAM-F-46-1	Infomasi yang dapat dilihat adalah alur penghargaan dan jenis penghargaan yang akan	Melihat Penghargaan Donor	Pendonor

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		diterima dari jumlah riwayat donor pendonor.		
47	SAM-F-47	Sistem harus mampu menampilkan informasi detail riwayat donor.	Melihat Riwayat Donor	Pendonor
47.1	SAM-F-47-1	Informasi yang dapat dilihat adalah tanggal donor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, riwayat penyakit, jumlah donor dan kesimpulan.		
48	SAM-F-48	Sistem harus mampu menyediakan sarana berupa formulir untuk menambahkan pengajuan permintaan darah.	Menambah Pengajuan Permintaan Darah	Petugas Rumah Sakit
48.1	SAM-F-48-1	Data formulir yang harus disediakan adalah nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, golongan darah dan file bukti surat dokter.		
49	SAM-F-49	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar pengajuan permintaan darah.	Melihat Daftar Pengajuan Permintaan Darah	Petugas Rumah Sakit
49.1	SAM-F-49-1	Data tabel yang disediakan adalah tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		komponen, nama pasien, dan nomor med.rec.		
50	SAM-F-50	Sistem harus mampu menampilkan informasi detail pengajuan permintaan darah berupa alur pengiriman darah dan detail permintaan.	Melihat Detail Pengajuan Permintaan Darah	Petugas Rumah Sakit
50.1	SAM-F-50-1	Informasi yang dapat dilihat untuk alur pengiriman darah adalah status darah saat ini, waktu, dan petugas. Dan untuk detail permintaan adalah tanggal permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan golongan darah.		
51	SAM-F-51	Sistem harus mampu menyediakan sarana untuk mencetak pengajuan permintaan darah.	Mencetak Pengajuan Permintaan Darah	Petugas Rumah Sakit
52	SAM-F-52	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar permintaan disetujui.	Melihat Daftar Permintaan Disetujui	Petugas Rumah Sakit
52.1	SAM-F-52-1	Data tabel yang disediakan adalah tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		komponen, nama pasien, dan nomor med.rec.		
53	SAM-F-53	Sistem harus mampu menampilkan informasi detail permintaan disetujui berupa alur pengiriman darah dan detail permintaan.		
53.1	SAM-F-53-1	Informasi yang dapat dilihat untuk alur pengiriman darah adalah status darah saat ini, waktu, dan petugas. Dan untuk detail permintaan adalah tanggal permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan golongan darah.	Melihat Detail Permintaan Disetujui	Petugas Rumah Sakit
54	SAM-F-54	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar darah diterima.		
54.1	SAM-F-54-1	Data tabel yang disediakan adalah tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.	Melihat Daftar Darah Diterima	Petugas Rumah Sakit
55	SAM-F-55	Sistem harus mampu menampilkan informasi detail darah diterima	Melihat Detail Darah Diterima	Petugas Rumah Sakit

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		berupa alur pengiriman darah dan detail permintaan.		
55.1	SAM-F-55-1	Informasi yang dapat dilihat untuk alur pengiriman darah adalah status darah saat ini, waktu, dan petugas. Dan untuk detail permintaan adalah tanggal permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan golongan darah.		
56	SAM-F-56	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar permintaan selesai.		
56.1	SAM-F-56-1	Data tabel yang disediakan adalah tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.	Melihat Daftar Permintaan Selesai	Petugas Rumah Sakit
57	SAM-F-57	Sistem harus mampu menampilkan informasi detail permintaan selesai berupa alur pengiriman darah dan detail permintaan.	Melihat Detail Permintaan Selesai	Petugas Rumah Sakit
57.1	SAM-F-57-1	Informasi yang dapat dilihat untuk alur		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		pengiriman darah adalah status darah saat ini, waktu, dan petugas. Dan untuk detail permintaan adalah tanggal permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan golongan darah.		
58	SAM-F-58	Sistem harus mampu menyediakan sarana untuk mencetak permintaan selesai.	Mencetak Permintaan Selesai	Petugas Rumah Sakit
59	SAM-F-59	Sistem harus mampu menyediakan sarana berupa tabel untuk melihat daftar permintaan dibatalkan.	Melihat Daftar Permintaan Dibatalkan	Petugas Rumah Sakit
59.1	SAM-F-59-1	Data tabel yang disediakan adalah tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.		
60	SAM-F-60	Sistem harus mampu menampilkan informasi detail permintaan dibatalkan berupa alur pengiriman darah dan detail permintaan.	Melihat Detail Permintaan Dibatalkan	Petugas Rumah Sakit
60.1	SAM-F-60-1	Infomasi yang dapat dilihat untuk alur pengiriman darah adalah		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		status darah saat ini, waktu, dan petugas. Dan untuk detail permintaan adalah tanggal permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan golongan darah.		
61	SAM-F-61	Sistem harus mampu menyediakan sarana untuk mencetak permintaan dibatalkan.	Mencetak Permintaan Dibatalkan	Petugas Rumah Sakit
62	SAM-F-62	Sistem harus mampu menampilkan informasi syarat permintaan darah.	Melihat Syarat Permintaan Darah	Petugas Rumah Sakit
62.1	SAM-F-62-1	Informasi yang dapat dilihat adalah dalam bentuk poin-poin yang menjelaskan syarat permintaan darah.		
63	SAM-F-63	Sistem harus mampu menampilkan informasi alur permintaan.	Melihat Informasi Alur Permintaan	Petugas Rumah Sakit
63.1	SAM-F-63-1	Informasi yang dapat dilihat adalah dalam bentuk poin-poin yang menjelaskan alur permintaan.		
64	SAM-F-64	Sistem harus mampu menampilkan informasi detail profil Petugas Rumah Sakit.	Melihat Profil Petugas Rumah Sakit	Petugas Rumah Sakit
64.1	SAM-F-64-1	Infomasi yang dapat dilihat adalah kode		

No.	Kode	Kebutuhan Fungsional	Use Case	Aktor
		petugas, username, nama petugas, nama rumah sakit, alamat rumah sakit, dan telepon rumah sakit.		
65	SAM-F-65	Sistem harus mampu menyediakan sarana untuk melakukan perubahan pada profil Petugas Rumah Sakit.	Mengubah Profil Petugas Rumah Sakit	Petugas Rumah Sakit
65.1	SAM-F-65-1	Data yang dapat diubah adalah kode petugas, username, nama petugas, nama rumah sakit, alamat rumah sakit, dan telepon rumah sakit.		

Tabel 4.3 Kebutuhan Non-Fungsional

No.	Kode	Spesifikasi	Parameter
1	SAM-NF-01	Sistem dapat dijalankan pada browser yang berbeda.	<i>Compatibility</i>

4.5 Pemodelan Kebutuhan Sistem

Pemodelan Kebutuhan digunakan untuk memudahkan proses pembacaan kebutuhan pada tahap perancangan. Dalam pemodelan kebutuhan ini dispesifikasikan pada suatu *use case diagram*. Setiap *use case* yang dimodelkan akan dijelaskan lebih detail pada bagian *use case scenario*.

4.5.1 Use Case Diagram

Use case diagram digunakan untuk mengetahui aktor dan apa yang dilakukan aktor tersebut pada sistem. Aktor-aktor yang berinteraksi di dalam sistem dimodelkan ke dalam bentuk *use case diagram* berdasarkan hasil analisis identifikasi aktor. Berdasarkan tabel maka dapat dibentuk sebuah *use case diagram* dengan tujuan menggambarkan keterlibatan dan interaksi aktor di dalam sistem. *Use Case Diagram* Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD) PMI Malang digambarkan pada Gambar 4.1.



4.5.2 Use Case Scenario

Use case scenario menjabarkan alur kinerja setiap *use case* melalui *scenario*. *Scenario* yang telah digambarkan pada masing-masing *use case* pada Gambar 4.1 sebelumnya, mengenai *use case diagram* sistem aplikasi manajemen unit transfusi darah.

Tabel 4.4 Use Case Scenario Pendaftaran

Pendaftaran	
Objective	Use case ini menjelaskan aktivitas aktor melakukan pendaftaran.
Actors	<i>Guest</i>
Pre-condition	Aktor telah berada pada halaman formulir pendaftaran.
Main flow	<ol style="list-style-type: none"> 1. Aktor memasukan data aktor dengan benar dan memilih tombol "Daftar". 2. Sistem menyimpan data aktor yang telah dibuat.
Alternative flows	<ol style="list-style-type: none"> 1a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan". 1b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil mendaftarkan aktor ke dalam sistem.

Tabel 4.5 Use Case Scenario Login

Login	
Objective	Use case ini menjelaskan aktivitas aktor melakukan login untuk masuk ke dalam sistem.
Actors	<i>Guest</i>
Pre-condition	Aktor telah berada pada halaman formulir <i>login</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memasukan data dengan benar dan memilih tombol "Login". 2. Sistem menampilkan halaman awal sesuai dengan status akun aktor yang telah terotorisasi yaitu sebagai <i>administrator</i>, <i>pendonor</i>, atau petugas rumah sakit.
Alternative flows	<ol style="list-style-type: none"> 1a. Apabila <i>username</i> dan <i>password</i> tidak sesuai, maka aktor tidak dapat masuk ke dalam sistem dan sistem akan menampilkan pesan peringatan "<i>username</i> dan <i>password</i> tidak valid" dan meminta aktor untuk memasukkan <i>username</i> dan <i>password</i> kembali. 1b. Apabila aktor hanya mengisi kolom <i>username</i> atau <i>password</i> dan memilih tombol "Login" maka sistem

	akan menampilkan pesan peringatan " <i>please fill out this field</i> ".
Post-condition	Aktor terotorisasi dan dapat mengakses sistem sebagai <i>administrator</i> , pendonor, atau petugas rumah sakit.

Tabel 4.6 Use Case Scenario Logout

Logout	
Objective	Use case ini menjelaskan aktivitas aktor melakukan <i>logout</i> .
Actors	Petugas Rumah Sakit, Pendonor, <i>Administrator</i>
Pre-condition	Aktor telah terotorisasi dan masuk ke dalam sistem.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>logout</i>. 2. Sistem menghapus <i>session</i> dan menampilkan halaman utama web.
Alternative flows	-
Post-condition	Aktor berhasil melakukan <i>logout</i>

Tabel 4.7 Use Case Scenario Melihat Stok Darah

Melihat Melihat Stok Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat stok darah.
Actors	<i>Guest dan Administrator</i>
Pre-condition	Aktor telah berada pada halaman web UTD PMI Kota Malang.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu beranda. 2. Sistem menampilkan tabel stok darah yang berisi nama produk, keterangan produk, dan jumlah darah pada semua golongan darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan tabel stok darah.

Tabel 4.8 Use Case Scenario Melihat Alur Donor

Melihat Alur Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat alur donor.
Actors	<i>Guest</i>
Pre-condition	Aktor telah berada pada halaman web UTD PMI Kota Malang.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu informasi.

	2. Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan poin-poin yang menjelaskan alur donor.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan halaman informasi alur donor.

Tabel 4.9 Use Case Scenario Melihat Informasi Setelah Donor

Melihat Informasi Setelah Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat informasi setelah donor.
Actors	<i>Guest</i>
Pre-condition	Aktor telah berada pada halaman web UTD PMI Kota Malang.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu informasi. 2. Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan poin-poin yang menjelaskan apa saja yang harus dilakukan setelah melakukan donor darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan halaman informasi setelah donor.

Tabel 4.10 Use Case Scenario Menambah Pesan Hubungi

Menambah Pesan Hubungi	
Objective	Use case ini menjelaskan aktivitas aktor saat akan melakukan menambah pesan hubungi.
Actors	<i>Guest</i>
Pre-condition	Aktor telah berada pada halaman web UTD PMI Kota Malang.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu hubungi kami. 2. Sistem akan menampilkan halaman hubungi kami. Halaman tersebut menampilkan formulir yang berisi nama lengkap, email, nomor telepon dan pesan yang akan dikirimkan. 3. Aktor mengisi formulir yang tersedia dan memilih tombol "submit".

	4. Sistem akan menyimpan data yang telah diisi oleh aktor.
Alternative flows	3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa <i>"please fill out this field"</i> .
Post-condition	Sistem berhasil menyimpan pesan hubungi.

Tabel 4.11 Use Case Scenario Melihat Daftar Pengguna

Melihat Daftar Pengguna	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar pengguna.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terotorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu data user. 2. Sistem akan menampilkan halaman daftar pengguna berupa tabel dengan data berupa <i>username</i> dan level pengguna.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar pengguna.

Tabel 4.12 Use Case Scenario Menambah Stok Darah

Menambah Stok Darah	
Objective	Use case ini menjelaskan aktivitas aktor menambah stok darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman stok darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol "tambah stok darah". 2. Sistem menampilkan formulir stok darah. 3. Aktor mengisi formulir yang berisi produk, golongan darah, jumlah stok darah, dan memilih tombol "Submit". 4. Sistem akan menyimpan data.
Alternative flows	3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Post-condition	Sistem berhasil menambah stok darah dan menampilkan halaman stok darah.

Tabel 4.13 *Use Case Scenario* Menambah Pengguna

Menambah Pengguna	
Objective	Use case ini menjelaskan aktivitas aktor menambah pengguna.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar pengguna.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengguna”. 2. Sistem menampilkan formulir pada halaman tambah pengguna. 3. Aktor mengisi formulir yang berisi <i>username</i>, <i>password</i>, <i>level</i>, dan memilih tombol “Submit”. 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan pengguna.

Tabel 4.14 *Use Case Scenario* Mengubah Data Pengguna

Mengubah Data Pengguna	
Objective	Use case ini menjelaskan aktivitas aktor mengubah data pengguna.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar pengguna.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan diubah dan memilih tombol “edit”. 2. Sistem akan menampilkan formulir yang telah terisi dengan data masukkan sebelumnya. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol “submit”. 4. Sistem menyimpan perubahan data pengguna.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan data pengguna dan menampilkan halaman daftar pengguna.

Tabel 4.15 *Use Case Scenario* Menghapus Pengguna

Menghapus Pengguna	
Objective	Use case ini menjelaskan aktivitas aktor menghapus pengguna.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar pengguna.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dihapus dari daftar pengguna dan memilih tombol “hapus”. 2. Sistem menampilkan pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “ok”. 4. Sistem menghapus pengguna yang dipilih aktor.
Alternative flows	3a. Apabila aktor memilih tombol “cancel” pada pesan konfirmasi maka proses penghapusan akan dibatalkan.
Post-condition	Sistem berhasil menghapus pengguna dan menampilkan halaman daftar pengguna.

Tabel 4.16 *Use Case Scenario* Melihat Daftar Permintaan Darah

Melihat Daftar Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar permintaan darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terotorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu petugas dan memilih sub menu permintaan darah. 2. Sistem akan menampilkan halaman daftar permintaan darah berupa tabel dengan data berupa tanggal, status, nama rumah sakit, nomor permintaan, golongan darah, jumlah komponen, dan jenis komponen.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar permintaan darah.

Tabel 4.17 *Use Case Scenario* Menambah Permintaan Darah

Menambah Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor menambah permintaan darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah permintaan”. 2. Sistem menampilkan formulir pada halaman formulir permintaan. 3. Aktor mengisi formulir yang benar dan lengkap serta melakukan upload file dan memilih tombol “Submit”. 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan. 3c. Menampilkan pesan error apabila <i>upload</i> file gagal atau file tidak sesuai dengan peraturan yang telah diberitahukan.
Post-condition	Sistem berhasil menyimpan permintaan darah.

Tabel 4.18 *Use Case Scenario* Menambah Alur Pengiriman Darah

Menambah Alur Pengiriman Darah	
Objective	Use case ini menjelaskan aktivitas aktor menambah alur pengiriman darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan ditambah alur pengiriman darah dan memilih tombol “tambah”. 2. Sistem menampilkan formulir pada halaman formulir alur pengiriman darah. 3. Aktor mengisi formulir yang benar dan lengkap dan memilih tombol “Submit”. 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Menampilkan status disetujui apabila status alur pengiriman bernilai permintaan disetujui.

	<p>3c. Menampilkan status diterima apabila status alur pengiriman bernilai darah diterima rs.</p> <p>3d. Menampilkan status selesai apabila status alur pengiriman bernilai permintaan selesai.</p> <p>3e. Menampilkan status dibatalkan apabila status alur pengiriman bernilai permintaan dibatalkan atau permintaan dibatalkan dengan alasan.</p>
Post-condition	Sistem berhasil menyimpan alur pengiriman darah dan menampilkan halaman daftar permintaan darah.

Tabel 4.19 *Use Case Scenario* Melihat Detail Permintaan Darah

Melihat Detail Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail permintaan darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol "lihat". 2. Sistem akan menampilkan halaman detail permintaan darah dengan data berupa waktu permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur pasien, agama pasien, jenis kelamin, dan file bukti surat.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail permintaan darah.

Tabel 4.20 *Use Case Scenario* Mengubah Data Permintaan Darah

Mengubah Data Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor mengubah data permintaan darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan diubah dan memilih tombol "edit". 2. Sistem akan menampilkan formulir yang telah terisi dengan data masukkan sebelumnya. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol "submit".

	4. Sistem menyimpan perubahan data permintaan darah.
Alternative flows	3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan data permintaan darah dan menampilkan halaman daftar permintaan darah.

Tabel 4.21 Use Case Scenario Melihat Daftar Data Petugas

Melihat Daftar Data Petugas	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar data petugas.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terotorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu petugas dan memilih sub menu data petugas. 2. Sistem akan menampilkan halaman daftar data petugas berupa tabel dengan data berupa nama rumah sakit, nama petugas, alamat rumah sakit, dan telepon rumah sakit.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar data petugas.

Tabel 4.22 Use Case Scenario Mengubah Data Petugas

Mengubah Data Petugas	
Objective	Use case ini menjelaskan aktivitas aktor mengubah data petugas.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar data petugas.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan diubah dan memilih tombol “<i>edit</i>”. 2. Sistem akan menampilkan formulir yang telah terisi dengan data masukkan sebelumnya. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol “<i>submit</i>”. 4. Sistem menyimpan perubahan data petugas.

Alternative flows	<p>3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.</p> <p>3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.</p>
Post-condition	Sistem berhasil menyimpan data petugas dan menampilkan halaman daftar data petugas.

Tabel 4.23 Use Case Scenario Menghapus Data Petugas

Menghapus Data Petugas	
Objective	Use case ini menjelaskan aktivitas aktor menghapus data petugas.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar data petugas.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dihapus dari daftar data petugas dan memilih tombol “hapus”. 2. Sistem menampilkan pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “ok”. 4. Sistem menghapus data petugas yang dipilih aktor.
Alternative flows	3a. Apabila aktor memilih tombol “cancel” pada pesan konfirmasi maka proses penghapusan akan dibatalkan.
Post-condition	Sistem berhasil menghapus data petugas dan menampilkan halaman daftar data petugas.

Tabel 4.24 Use Case Scenario Melihat Daftar Pengajuan Donor

Melihat Daftar Pengajuan Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar pengajuan donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terautorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor dan memilih sub menu pengajuan donor. 2. Sistem akan menampilkan halaman daftar pengajuan donor berupa tabel dengan data berupa tanggal, nomor pengajuan, kode pendonor, nama pendonor, dan golongan darah.

Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar pengajuan donor.

Tabel 4.25 *Use Case Scenario* Menambah Pengajuan Donor

Menambah Pengajuan Donor	
Objective	Use case ini menjelaskan aktivitas aktor menambah pengajuan donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar pengajuan donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengajuan”. 2. Sistem menampilkan formulir pada halaman formulir pengajuan. 3. Aktor mengisi formulir yang benar dan lengkap dan memilih tombol “Submit”. 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan pengajuan donor.

Tabel 4.26 *Use Case Scenario* Melihat Detail Pengajuan Donor

Melihat Detail Pengajuan Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail pengajuan donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar pengajuan donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol “lihat”. 2. Sistem akan menampilkan halaman detail pengajuan donor dengan data berupa tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur pendonor, golongan darah, alamat pendonor, telepon pendonor.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail pengajuan donor.

Tabel 4.27 Use Case Scenario Menghapus Pengajuan Donor

Menghapus Pengajuan Donor	
Objective	Use case ini menjelaskan aktivitas aktor menghapus pengajuan donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar pengajuan donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dihapus dari daftar pengajuan donor dan memilih tombol “hapus”. 2. Sistem menampilkan pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “ok”. 4. Sistem menghapus pengajuan donor yang dipilih aktor.
Alternative flows	3a. Apabila aktor memilih tombol “cancel” pada pesan konfirmasi maka proses penghapusan akan dibatalkan.
Post-condition	Sistem berhasil menghapus pengajuan donor dan menampilkan halaman daftar pengajuan donor.

Tabel 4.28 Use Case Scenario Melihat Daftar Data Pendonor

Melihat Daftar Data Pendonor	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar data pendonor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terotorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor dan memilih sub menu data pendonor. 2. Sistem akan menampilkan halaman daftar data pendonor berupa tabel dengan data berupa kode pendonor, nomor identitas, nama pendonor, umur, dan telepon.
Alternative flows	-
Post-condition	Sistem berhasil melihat daftar data pendonor.

Tabel 4.29 *Use Case Scenario* Melihat Detail Data Pendonor

Melihat Detail Data Pendonor	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail data pendonor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar data pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol “lihat”. 2. Sistem akan menampilkan halaman detail data pendonor dengan data berupa kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail data pendonor.

Tabel 4.30 *Use Case Scenario* Mengubah Data Pendonor

Mengubah Data Pendonor	
Objective	Use case ini menjelaskan aktivitas aktor mengubah data pendonor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar data pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan diubah dan memilih tombol “edit”. 2. Sistem akan menampilkan formulir yang telah terisi dengan data masukkan sebelumnya. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol “submit”. 4. Sistem menyimpan perubahan data pendonor.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan data pendonor dan menampilkan halaman daftar data pendonor.

Tabel 4.31 *Use Case Scenario* Menghapus Data Pendoron

Menghapus Data Pendoron	
Objective	Use case ini menjelaskan aktivitas aktor menghapus data pendonor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar data pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dihapus dari daftar data pendonor dan memilih tombol “hapus”. 2. Sistem menampilkan pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “ok”. 4. Sistem menghapus data pendonor yang dipilih aktor.
Alternative flows	3a. Apabila aktor memilih tombol “cancel” pada pesan konfirmasi maka proses penghapusan akan dibatalkan.
Post-condition	Sistem berhasil menghapus data pendonor dan menampilkan halaman daftar data pendonor.

Tabel 4.32 *Use Case Scenario* Melihat Daftar Riwayat Donor Pendoron

Melihat Daftar Riwayat Donor Pendoron	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar riwayat donor pendonor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terotorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor dan memilih sub menu riwayat donor. 2. Sistem akan menampilkan halaman daftar riwayat donor berupa tabel dengan data berupa tanggal donor, kode pendonor, status kesehatan, riwayat donor, dan status riwayat penyakit.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar riwayat donor.

Tabel 4.33 *Use Case Scenario* Menambah Riwayat Donor

Menambah Riwayat Donor	
Objective	Use case ini menjelaskan aktivitas aktor menambah riwayat donor.
Actors	<i>Administrator</i>

Pre-condition	Aktor telah berada pada halaman daftar riwayat donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah riwayat donor”. 2. Sistem menampilkan formulir pada halaman formulir riwayat donor. 3. Aktor mengisi formulir dengan benar dan lengkap dan memilih tombol “Submit”. 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan riwayat donor.

Tabel 4.34 Use Case Scenario Melihat Detail Riwayat Donor

Melihat Detail Riwayat Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail riwayat donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar riwayat donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol “lihat”. 2. Sistem akan menampilkan halaman detail riwayat donor dengan data berupa waktu donor, kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail riwayat donor.

Tabel 4.35 Use Case Scenario Mengubah Data Riwayat Donor

Mengubah Data Riwayat Donor	
Objective	Use case ini menjelaskan aktivitas aktor mengubah data riwayat donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar riwayat donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan diubah dan memilih tombol “edit”.

	<ol style="list-style-type: none"> 2. Sistem akan menampilkan formulir yang telah terisi dengan data masukkan sebelumnya. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol "submit". 4. Sistem menyimpan perubahan data riwayat donor.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan". 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan data riwayat donor dan menampilkan halaman daftar riwayat donor pendonor.

Tabel 4.36 Use Case Scenario Menghapus Riwayat Donor

Menghapus Riwayat Donor	
Objective	Use case ini menjelaskan aktivitas aktor menghapus riwayat donor.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar riwayat donor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dihapus dari daftar riwayat donor dan memilih tombol "hapus". 2. Sistem menampilkan pesan konfirmasi "apakah anda yakin menghapus data?". 3. Aktor memilih tombol "ok". 4. Sistem menghapus riwayat donor yang dipilih aktor.
Alternative flows	3a. Apabila aktor memilih tombol "cancel" pada pesan konfirmasi maka proses penghapusan akan dibatalkan.
Post-condition	Sistem berhasil menghapus riwayat donor.

Tabel 4.37 Use Case Scenario Melihat Daftar Data Butuh Darah

Melihat Daftar Data Butuh Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar data butuh darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terautorisasi sebagai <i>administrator</i> .
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor dan memilih sub menu butuh darah. 2. Sistem akan menampilkan halaman daftar butuh darah berupa tabel dengan data berupa tanggal, nama wali,

	nomor telepon, golongan darah, faktor rhesus(Rh), jumlah komponen, dan rumah sakit.
Alternative flows	-
Post-condition	Aktor berhasil melihat daftar butuh darah.

Tabel 4.38 *Use Case Scenario* Menambah Data Butuh Darah

Menambah Data Butuh Darah	
Objective	Use case ini menjelaskan aktivitas aktor menambah data butuh darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar data butuh darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol "tambah informasi". 2. Sistem menampilkan formulir pada halaman formulir tambah data. 3. Aktor mengisi formulir dengan benar dan lengkap dan memilih tombol "Submit". 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan". 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan butuh darah.

Tabel 4.39 *Use Case Scenario* Melihat Detail Butuh Darah

Melihat Detail Butuh Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail butuh darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar butuh darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol "lihat". 2. Sistem akan menampilkan halaman detail butuh darah dengan data berupa tanggal mengajukan, nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.
Alternative flows	-

Post-condition	Sistem berhasil menampilkan detail butuh darah.
-----------------------	---

Tabel 4.40 Use Case Scenario Mengubah Data Butuh Darah

Mengubah Data Butuh Darah	
Objective	Use case ini menjelaskan aktivitas aktor mengubah data butuh darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar butuh darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan diubah dan memilih tombol "edit". 2. Sistem akan menampilkan formulir yang telah terisi dengan data masukkan sebelumnya. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol "submit". 4. Sistem menyimpan perubahan data butuh darah.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan". 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan data butuh darah dan menampilkan halaman daftar butuh darah.

Tabel 4.41 Use Case Scenario Menghapus Data Butuh Darah

Menghapus Data Butuh Darah	
Objective	Use case ini menjelaskan aktivitas aktor menghapus butuh darah.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah berada pada halaman daftar butuh darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dihapus dari daftar butuh darah dan memilih tombol "hapus". 2. Sistem menampilkan pesan konfirmasi "apakah anda yakin menghapus data?". 3. Aktor memilih tombol "ok". 4. Sistem menghapus butuh darah yang dipilih aktor.
Alternative flows	3a. Apabila aktor memilih tombol "cancel" pada pesan konfirmasi maka proses penghapusan akan dibatalkan.
Post-condition	Sistem berhasil menghapus butuh darah dan menampilkan halaman daftar butuh darah.

Tabel 4.42 *Use Case Scenario* Melihat Pemberitahuan

Melihat Pemberitahuan	
Objective	Use case ini menjelaskan aktivitas aktor melihat pemberitahuan.
Actors	<i>Administrator</i>
Pre-condition	Aktor telah menekan tombol “lonceng” pemberitahuan.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan menekan data tersebut. 2. Sistem akan menampilkan halaman daftar permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan pemberitahuan baru yang ada.

Tabel 4.43 *Use Case Scenario* Melihat Syarat Donor

Melihat Syarat Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat syarat donor.
Actors	Pendonor
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor berada pada halaman pendonor. 2. Sistem menampilkan poin-poin yang menjelaskan syarat-syarat donor.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan syarat donor.

Tabel 4.44 *Use Case Scenario* Menambah Pengajuan Donor

Menambah Pengajuan Donor	
Objective	Use case ini menjelaskan aktivitas aktor menambah pengajuan donor.
Actors	Pendonor
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Pengajuan Donor”. 2. Sistem menampilkan formulir pengajuan donor.

	<p>3. Aktor mengecek kebenaran data dan mengisi formulir pengajuan yang berisi data diri dan data pengajuan donor lalu menekan tombol “Submit”.</p> <p>4. Sistem akan menyimpan data.</p>
Alternative flows	3a. Salah satu data belum terisi dengan benar atau kosong maka akan terdapat pesan ‘Bidang tersebut dibutuhkan’.
Post-condition	Sistem berhasil menyimpan pengajuan donor.

Tabel 4.45 Use Case Scenario Mencetak Pengajuan Donor

Mencetak Pengajuan Donor	
Objective	Use case ini menjelaskan aktivitas aktor mencetak pengajuan donor.
Actors	Pendonor
Pre-condition	Sistem menampilkan halaman hasil formulir yang telah diisi aktor.
Main flow	<p>1. Aktor memilih tombol “print”.</p> <p>2. Sistem akan menampilkan format formulir pengajuan donor untuk dicetak.</p> <p>3. Aktor memilih tombol “print”.</p> <p>4. Sistem mencetak formulir pengajuan donor.</p>
Alternative flows	-
Post-condition	Sistem berhasil mencetak pengajuan donor.

Tabel 4.46 Use Case Scenario Melihat Informasi Butuh Darah

Melihat Informasi Butuh Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat informasi butuh darah.
Actors	Pendonor
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai pendonor.
Main flow	<p>1. Aktor berada pada halaman utama pendonor dan memilih menu butuh donor darah.</p> <p>2. Sistem akan menampilkan halaman informasi butuh donor darah sesuai dengan golongan darah aktor yang telah terotorisasi masuk ke dalam sistem.</p>
Alternative flows	3a. Menampilkan data golongan darah O+, A+, B+, dan AB+ apabila golongan darah pendonor yang telah terotorisasi bergolongan darah O+.

	<p>3b. Menampilkan data golongan darah A+ dan AB+ apabila golongan darah pendonor yang telah terotorisasi bergolongan darah A+.</p> <p>3c. Menampilkan data golongan darah B+ dan AB+ apabila golongan darah pendonor yang telah terotorisasi bergolongan darah B+.</p> <p>3d. Menampilkan data golongan darah AB+ apabila golongan darah pendonor yang telah terotorisasi bergolongan darah AB+.</p> <p>3e. Menampilkan data golongan darah O+, A+, B+, AB+, O-, A-, B-, dan AB- apabila golongan darah pendonor yang telah terotorisasi bergolongan darah O-.</p> <p>3f. Menampilkan data golongan darah A+, AB+, A- dan AB- apabila golongan darah pendonor yang telah terotorisasi bergolongan darah A-.</p> <p>3g. Menampilkan data golongan darah B+, AB+, B- dan AB- apabila golongan darah pendonor yang telah terotorisasi bergolongan darah B-.</p> <p>3h. Menampilkan data golongan darah AB+ dan AB- apabila golongan darah pendonor yang telah terotorisasi bergolongan darah AB-.</p>
Post-condition	Sistem berhasil menampilkan informasi butuh darah.

Tabel 4.47 Use Case Scenario Melihat Profil Pendonor

Melihat Profil Pendonor	
Objective	Use case ini menjelaskan aktivitas aktor melihat profil pendonor.
Actors	Pendonor
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "<i>setting</i> profil". 2. Sistem akan menampilkan halaman <i>setting</i> profil pada tab profil yang berisi kode pendonor, username, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan profil pendonor.

Tabel 4.48 Use Case Scenario Mengubah Profil Pendoror

Mengubah Profil Pendoror	
Objective	Use case ini menjelaskan aktivitas aktor mengubah profil pendonor.
Actors	Pendonor
Pre-condition	Aktor telah berada pada halaman <i>setting</i> profil.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol "<i>edit data</i>". 2. Sistem akan menampilkan formulir yang telah terisi dengan data profil aktor. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol "<i>submit</i>". 4. Sistem menyimpan perubahan data profil pendonor.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan". 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan profil pendonor dan menampilkan halaman setting profil.

Tabel 4.49 Use Case Scenario Melihat Penghargaan Donor

Melihat Penghargaan Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat penghargaan donor.
Actors	Pendonor
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "Riwayat Donor". 2. Sistem menampilkan halaman riwayat donor dan menampilkan informasi penghargaan donor yang berisi alur penghargaan dan jenis penghargaan yang akan diterima dari jumlah riwayat donor pendonor.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan penghargaan donor.

Tabel 4.50 *Use Case Scenario* Melihat Riwayat Donor

Melihat Riwayat Donor	
Objective	Use case ini menjelaskan aktivitas aktor melihat riwayat donor.
Actors	Pendonor
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai pendonor.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu “Riwayat Donor”. 2. Sistem menampilkan halaman riwayat donor dan menampilkan tabel informasi riwayat donor yang berisi tanggal donor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, riwayat penyakit, jumlah donor dan kesimpulan.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan riwayat donor.

Tabel 4.51 *Use Case Scenario* Menambah Pengajuan Permintaan Darah

Menambah Pengajuan Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor menambah pengajuan permintaan darah.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “tambah Permintaan”. 2. Sistem menampilkan formulir permintaan darah. 3. Aktor mengisi formulir yang benar dan lengkap serta melakukan upload file dan memilih tombol “Submit”. 4. Sistem akan menyimpan data.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”. 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan. 3c. Menampilkan pesan error apabila <i>upload</i> file gagal atau file tidak sesuai dengan peraturan yang telah diberitahukan.
Post-condition	Sistem berhasil menyimpan pengajuan permintaan darah.

Tabel 4.52 *Use Case Scenario* Melihat Daftar Pengajuan Permintaan Darah

Melihat Daftar Pengajuan Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar pengajuan permintaan darah.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah masuk ke halaman utama sistem dan terotorisasi sebagai petugas rumah sakit.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu permintaan darah dan memilih tab menu menunggu persetujuan. 2. Sistem akan menampilkan daftar permintaan darah pada tab menu menunggu persetujuan berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar pengajuan permintaan darah.

Tabel 4.53 *Use Case Scenario* Melihat Detail Pengajuan Permintaan Darah

Melihat Detail Pengajuan Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail pengajuan permintaan darah.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu menunggu persetujuan.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol "lihat". 2. Sistem akan menampilkan halaman detail permintaan darah pada tab menu menunggu persetujuan yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail pengajuan permintaan darah.

Tabel 4.54 *Use Case Scenario* Mencetak Pengajuan Permintaan Darah

Mencetak Pengajuan Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor mencetak pengajuan permintaan darah.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu menunggu persetujuan.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dicetak dan menekan tombol “cetak”. 2. Sistem akan menampilkan format formulir permintaan darah untuk dicetak. 3. Aktor memilih tombol “print”. 4. Sistem mencetak formulir permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil mencetak pengajuan permintaan darah.

Tabel 4.55 *Use Case Scenario* Melihat Daftar Permintaan Disetujui

Melihat Daftar Permintaan Disetujui	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar permintaan disetujui.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tab menu permintaan disetujui. 2. Sistem akan menampilkan daftar permintaan darah pada tab menu permintaan disetujui berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar permintaan disetujui.

Tabel 4.56 *Use Case Scenario* Melihat Detail Permintaan Disetujui

Melihat Detail Permintaan Disetujui	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail permintaan disetujui.
Actors	Petugas Rumah Sakit

Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu permintaan disetujui.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol “lihat”. 2. Sistem akan menampilkan halaman detail permintaan darah yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail permintaan disetujui.

Tabel 4.57 Use Case Scenario Melihat Daftar Darah Diterima

Melihat Daftar Darah Diterima	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar darah diterima.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tab menu darah diterima. 2. Sistem akan menampilkan daftar permintaan darah pada tab menu darah diterima berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar darah diterima.

Tabel 4.58 Use Case Scenario Melihat Detail Darah Diterima

Melihat Detail Darah Diterima	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail darah diterima.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu darah diterima.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol “lihat”. 2. Sistem akan menampilkan halaman detail permintaan darah pada tab menu darah diterima yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.

Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail darah diterima.

Tabel 4.59 *Use Case Scenario* Melihat Daftar Permintaan Selesai

Melihat Daftar Permintaan Selesai	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar permintaan selesai.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tab menu pesan selesai. 2. Sistem akan menampilkan daftar permintaan darah pada tab menu pesan selesai berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar permintaan selesai.

Tabel 4.60 *Use Case Scenario* Melihat Detail Permintaan Selesai

Melihat Detail Permintaan Selesai	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail permintaan selesai.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu pesan selesai.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol "lihat". 2. Sistem akan menampilkan halaman detail permintaan darah pada tab menu menunggu persetujuan yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail permintaan selesai.

Tabel 4.61 *Use Case Scenario* Mencetak Permintaan Selesai

Mencetak Permintaan Selesai	
Objective	Use case ini menjelaskan aktivitas aktor mencetak permintaan selesai.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu pesan selesai.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dicetak dan menekan tombol “cetak”. 2. Sistem akan menampilkan format formulir permintaan darah untuk dicetak. 3. Aktor memilih tombol “print”. 4. Sistem mencetak formulir permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil mencetak permintaan selesai.

Tabel 4.62 *Use Case Scenario* Melihat Daftar Permintaan Dibatalkan

Melihat Daftar Permintaan Dibatalkan	
Objective	Use case ini menjelaskan aktivitas aktor melihat daftar permintaan dibatalkan.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman permintaan darah.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tab menu pesan batal. 2. Sistem akan menampilkan daftar permintaan darah pada tab menu pesan batal berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan daftar permintaan dibatalkan.

Tabel 4.63 *Use Case Scenario* Melihat Detail Permintaan Dibatalkan

Melihat Detail Permintaan Dibatalkan	
Objective	Use case ini menjelaskan aktivitas aktor melihat detail permintaan dibatalkan.
Actors	Petugas Rumah Sakit

Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu pesan batal.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dilihat dan memilih tombol “lihat”. 2. Sistem akan menampilkan halaman detail permintaan darah pada tab menu menunggu persetujuan yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan detail permintaan dibatalkan.

Tabel 4.64 *Use Case Scenario* Mencetak Permintaan Dibatalkan

Mencetak Permintaan Dibatalkan	
Objective	Use case ini menjelaskan aktivitas aktor mencetak permintaan dibatalkan.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman daftar permintaan darah tab menu pesan batal.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih data yang akan dicetak dan menekan tombol “cetak”. 2. Sistem akan menampilkan format permintaan darah untuk dicetak. 3. Aktor memilih tombol “print”. 4. Sistem mencetak formulir permintaan darah.
Alternative flows	-
Post-condition	Sistem berhasil mencetak permintaan dibatalkan.

Tabel 4.65 *Use Case Scenario* Melihat Syarat Permintaan Darah

Melihat Syarat Permintaan Darah	
Objective	Use case ini menjelaskan aktivitas aktor melihat syarat permintaan darah.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai petugas rumah sakit.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu informasi. 2. Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan informasi poin-poin yang menjelaskan syarat permintaan darah.

Alternative flows	-
Post-condition	Sistem berhasil menampilkan syarat permintaan darah.

Tabel 4.66 *Use Case Scenario* Melihat Informasi Alur Permintaan

Melihat Informasi Alur Permintaan	
Objective	Use case ini menjelaskan aktivitas aktor melihat informasi alur permintaan.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai petugas rumah sakit.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih menu informasi. 2. Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan informasi poin-poin yang menjelaskan alur permintaan.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan informasi alur permintaan.

Tabel 4.67 *Use Case Scenario* Melihat Profil Petugas Rumah Sakit

Melihat Profil Petugas Rumah Sakit	
Objective	Use case ini menjelaskan aktivitas aktor melihat profil petugas rumah sakit.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah masuk ke dalam sistem dan terotorisasi sebagai petugas rumah sakit.
Main flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol "<i>edit profil</i>" pada <i>dropdown</i> nama petugas rumah sakit. 2. Sistem akan menampilkan halaman detail profil yang berisi kode petugas, username, nama petugas, nama rumah sakit, alamat rumah sakit, dan telepon rumah sakit.
Alternative flows	-
Post-condition	Sistem berhasil menampilkan profil petugas rumah sakit.

Tabel 4.68 *Use Case Scenario* Mengubah Profil Petugas Rumah Sakit

Mengubah Profil Petugas Rumah Sakit	
Objective	Use case ini menjelaskan aktivitas aktor mengubah profil petugas rumah sakit.
Actors	Petugas Rumah Sakit
Pre-condition	Aktor telah berada pada halaman profil petugas.
Main flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol "Edit profil". 2. Sistem menampilkan formulir yang telah berisi data profil aktor. 3. Aktor mengubah data sesuai yang diinginkan dengan benar dan memilih tombol "submit". 4. Sistem menyimpan perubahan data profil petugas.
Alternative flows	<ol style="list-style-type: none"> 3a. Salah satu data kosong maka akan terdapat pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan". 3b. Salah satu data belum terisi dengan benar maka akan terdapat pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Post-condition	Sistem berhasil menyimpan profil petugas rumah sakit dan menampilkan halaman profil petugas.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

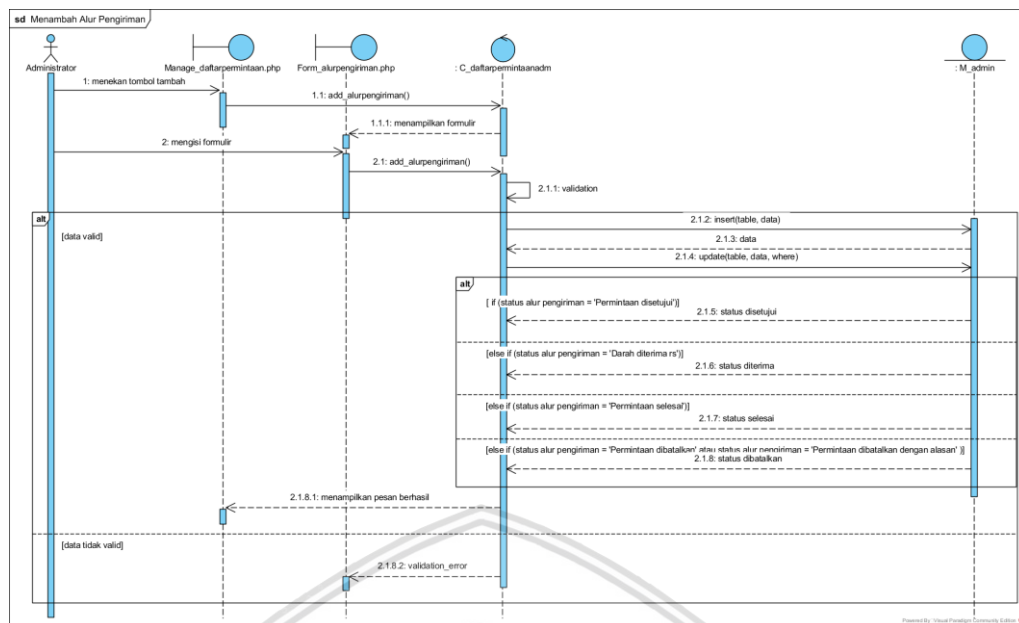
Perancangan memuat hal yang berhubungan dengan rancangan perangkat lunak yang akan dibangun dengan tujuan untuk mempermudah proses implementasi pada proses pembangunan sistem. Perancangan dilakukan dalam beberapa tahap yaitu pembuatan *sequence diagram*, perancangan basis data, pembuatan class diagram, perancangan algoritme, dan perancangan antarmuka.

5.1.1 *Sequence Diagram*

Sequence diagram adalah suatu diagram yang digunakan untuk mendefinisikan alur pertukaran pesan oleh objek-objek atau aktor yang melakukan suatu pekerjaan (Lethbridge, 2005). Pada perancangan ini dibuat 3 (tiga) sampel *sequence diagram*. Berikut ini merupakan *sequence diagram* yang telah dibuat. *Sequence diagram* mengubah alur pengiriman darah

5.1.1.1 *Sequence Diagram* Menambah Alur Pengiriman Darah

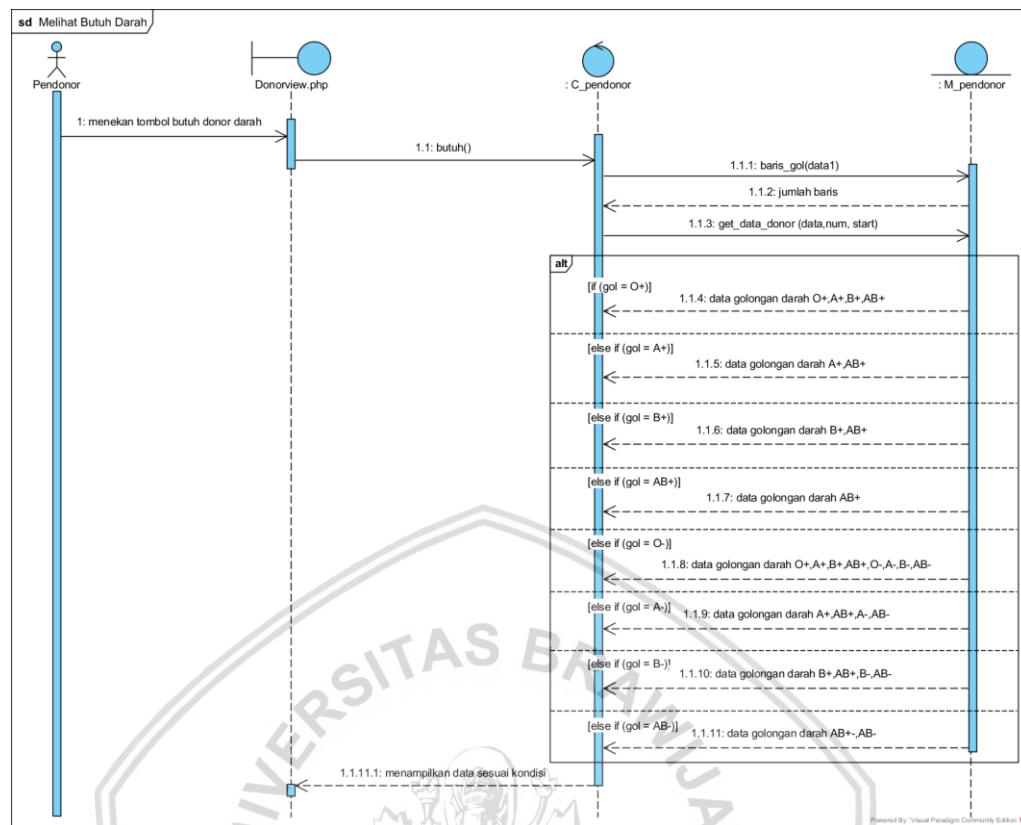
Sequence diagram menambah alur pengiriman darah diawali ketika *administrator* berada pada halaman daftar permintaan darah. *Administrator* menekan tombol tambah, kemudian sistem memproses melalui controller yang bernama `C_daftarpermintaanadm` dengan memanggil method `add_alurpengiriman` dan menampilkan formulir yang akan diisi oleh *administrator*. Setelah formulir diisi oleh *administrator* maka sistem akan memvalidasi data yang telah dimasukkan, jika data valid, maka sistem akan menyimpan data pada *database* lalu menampilkan pesan berhasil. Jika data tidak valid, maka sistem akan menampilkan validasi *error*. Alur tersebut direpresentasikan pada Gambar 5.1.



Gambar 5.1 Sequence Diagram Menambah Alur Pengiriman Darah

5.1.1.2 Sequence Diagram Melihat Informasi Butuh Darah

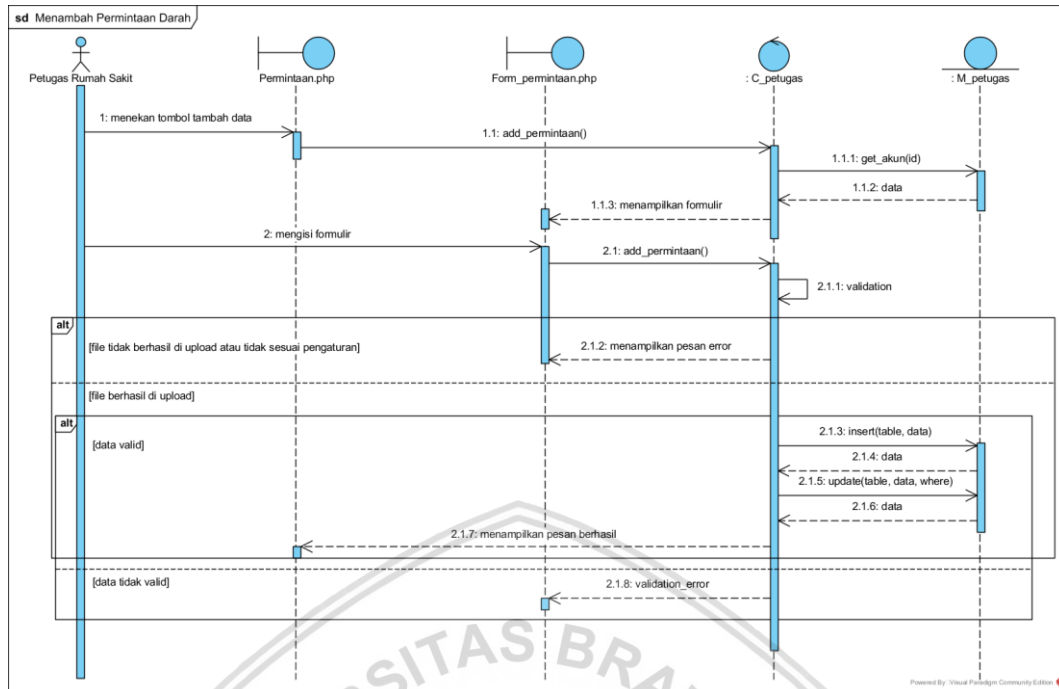
Sequence diagram melihat informasi butuh darah diawali ketika pendonor telah masuk ke halaman awal dan terotorisasi sebagai pendonor. Pendonor menekan tombol butuh donor darah, kemudian sistem memproses melalui controller yang bernama C_pendonor dengan memanggil method butuh. Lalu sistem akan memproses data yang akan ditampilkan berdasarkan kondisi yang dimiliki oleh pendonor. Alur tersebut direpresentasikan pada Gambar 5.2.



Gambar 5.2 Sequence Diagram Melihat informasi Butuh Darah

5.1.1.3 Sequence Diagram Menambah Permintaan Darah

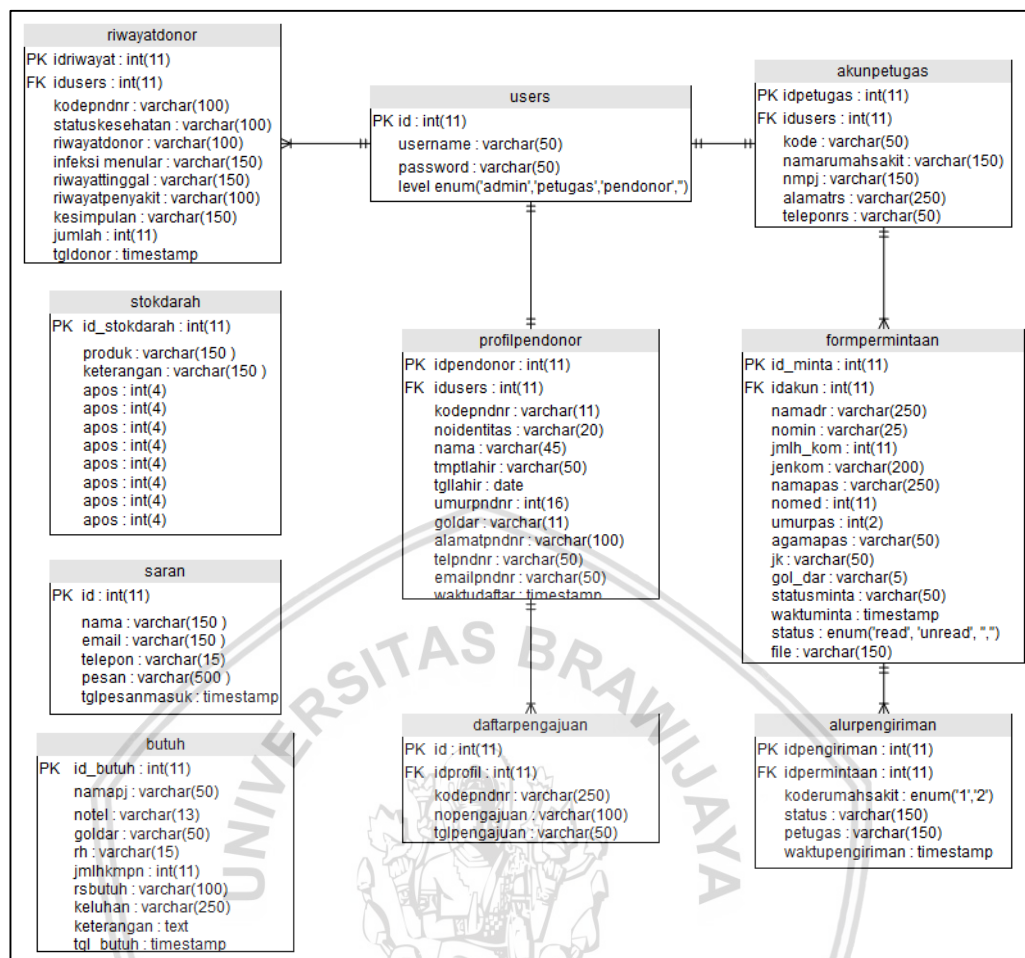
Sequence diagram menambah permintaan darah diawali ketika petugas rumah sakit berada pada halaman daftar permintaan darah. Petugas rumah sakit akan memilih menekan tombol tambah data, kemudian sistem memproses melalui controller yang bernama C_petugas dengan memanggil method add_permintaan() dan menampilkan formulir permintaan, kemudian petugas rumah sakit mengisi formulir, selanjutnya sistem akan memvalidasi data yang telah dimasukkan petugas rumah sakit, jika data valid, maka sistem akan menyimpan data pada database lalu menampilkan pesan berhasil. Jika data tidak valid, maka sistem akan menampilkan validasi error. Alur tersebut direpresentasikan pada Gambar 5.3.



Gambar 5.3 Sequence Diagram Menambah Permintaan Darah

5.1.2 Perancangan Basis Data

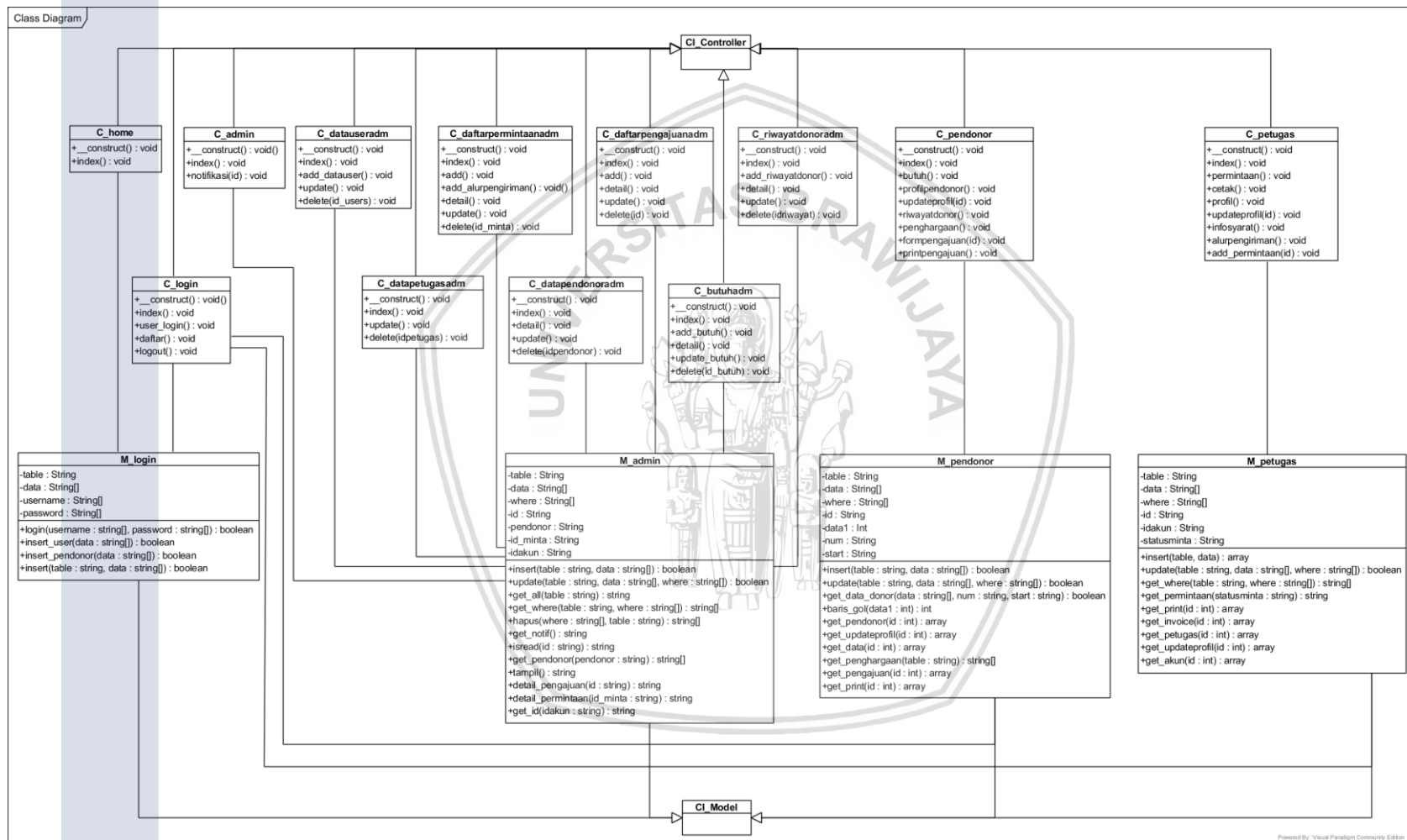
Perancangan ini bertujuan untuk memodelkan relasi antar entitas pada sistem. Perancangan dilakukan dengan cara perancangan fisik yang digambarkan dengan *Physical data model*. *Physical data model* digunakan untuk merepresentasikan rancangan basis data dan mendefinisikan semua struktur tabel, termasuk nama kolom, tipe data kolom, primary key, foreign key, dan relasi antar tabel. Gambar 5.5 merupakan rancangan *physical data model* sistem aplikasi manajemen UTD.



Gambar 5.4 Physical Data Model

5.1.3 Class Diagram

Perancangan *class diagram* bertujuan untuk mendefinisikan struktur klas pada sistem. Perancangan *class diagram* level perancangan pada sistem aplikasi manajemen Unit Transfusi Darah (UTD) dapat dilihat pada Gambar 5.4 pada Perancangan ini, struktur klas yang akan dibahas berupa klas-klas *Model*, *View*, dan *Controller* yang saling berelasi. Ketiga jenis klas tersebut dibedakan berdasarkan induk klasnya. Sistem aplikasi manajemen Unit Transfusi Darah (UTD) ini mengimplementasikan *Codeigniter*, terdapat klas *CI_Controller* dan *CI_Model* sebagai klas induk. Klas *controller* mempunyai hubungan generalisasi dengan klas *CI_Controller*, dan Klas *model* mempunyai hubungan generalisasi dengan klas *CI_Model*, dan klas *view* tidak mempunyai hubungan generalisasi dengan klas manapun. Klas *controller* diantaranya adalah klas *C_admin*, *C_butuhadm*, *C_daftarpermintaanadm*, *C_daftarpengajuanadm*, *C_datauseradm*, *C_datapetugasadm*, *C_datapendonoradm*, *C_riwayatdonoradm*, *C_home*, *C_login*, *C_pendonor*, dan *C_petugas*. Klas *model* *M_Admin*, *M_Login*, *M_pendonor*, dan *M_Petugas*.



Gambar 5.5 Class Diagram

5.1.4 Perancangan Algoritme

Pada bagian ini menjelaskan tentang perancangan algoritme dari sistem aplikasi manajemen Uinit Transfusi Darah (UTD). Perancangan algoritme dirancang berdasarkan perancangan *class diagram* yang telah dilakukan. Setiap *class* memiliki fungsi algoritme. Perancangan algoritme bertujuan untuk menjabarkan algoritme dari fungsi dari *class*. Didalam perancangan algoritme ini mencantumkan 3 algoritme berbeda dalam sistem.

5.1.4.1 Algoritme Menambah Alur Pengiriman Darah

Tabel 5.1 merupakan perancangan algoritme untuk menambah alur pengiriman darah. Algoritme ini akan diimplementasikan pada fungsi `add_alurpengiriman()` yang terdapat pada *class* `Daftarpermintaanadm`.

Tabel 5.1 Perancangan Algoritme Menambah Alur Pengiriman Darah

1	Mulai
2	IF (nilai input dari submit = true)
3	Menetapkan peraturan untuk validasi form
4	IF (validasi = true)
5	Inisialisasi array variabel alur pengiriman dengan inputan dari form
6	IF (status alur pengiriman = permintaan disetujui)
7	Inisialisasi variabel status minta adalah disetujui
8	ELSE IF (status alur pengiriman = darah diterima rs)
9	Inisialisasi variabel status minta adalah diterima
10	ELSE IF (status alur pengiriman = permintaan selesai)
11	Inisialisasi variabel status minta adalah selesai
12	ELSE IF (status alur pengiriman = permintaan dibatalkan atau permintaan dibatalkan dengan alasan)
13	Inisialisasi variabel status minta adalah dibatalkan
14	END IF
15	Memasukkan data ke dalam database
16	Memperbarui data dari database
17	Menampilkan pesan berhasil ditambah
18	ELSE
19	Menampilkan validasi error

20	END IF
21	ELSE
22	Menampilkan formulir menambah alur pengiriman
23	END IF
24	Selesai

5.1.4.2 *Algoritme* Melihat Informasi Butuh Darah

Tabel 5.2 merupakan perancangan algoritme untuk melihat informasi butuh darah. Algoritme ini akan diimplementasikan pada fungsi butuh() yang terdapat pada *class* pendonor.

Tabel 5.2 Perancangan *Algoritme* Melihat Informasi Butuh Darah

1	Mulai
2	Mengambil session golongan darah dari array
3	Inisialisasi variabel pendonor berupa array
4	IF (golongan darah = O+)
5	Inisialisasi variabel pendonor berupa O+, A+, B+, AB+
6	ELSE IF (golongan darah = A+)
7	Inisialisasi variabel pendonor berupa A+, AB+
8	ELSE IF (golongan darah = B+)
9	Inisialisasi variabel pendonor berupa B+, AB+
10	ELSE IF (golongan darah = AB+)
11	Inisialisasi variabel pendonor berupa AB+
12	ELSE IF (golongan darah = O-)
13	Inisialisasi variabel pendonor berupa O+, A+, B+, AB+, O-, A-, B-, AB-
14	ELSE IF (golongan darah = A-)
15	Inisialisasi variabel pendonor berupa A+, AB+, A-, AB-
16	ELSE IF (golongan darah = B-)
17	Inisialisasi variabel pendonor berupa B+, AB+, B-, AB-
18	ELSE IF (golongan darah = AB-)
19	Inisialisasi variabel pendonor berupa AB+, AB-
20	END IF
21	Mengambil jumlah baris data melalui model

22	Konfigurasi untuk settingan pagination
23	Mengambil data golongan darah sesuai kondisi dari database
24	Menampilkan data sesuai offset dan aturan per halaman yang ditentukan
25	Selesai

5.1.4.3 *Algoritme Menambah Permintaan Darah*

Tabel 5.3 merupakan perancangan algoritme untuk menambah permintaan darah. Algoritme ini akan diimplementasikan pada fungsi `add_permintaan()` yang terdapat pada *class* petugas.

Tabel 5.3 Perancangan *Algoritme Menambah Permintaan Darah*

Baris	Pseudocode
1	Mulai
2	IF (nilai input dari submit = true)
3	Menetapkan peraturan untuk validasi form
4	IF (validasi = true)
5	Inisialisasi array variabel petugas dengan inputan dari form
6	Inisialisasi array variabel formpermintaan dengan inputan dari form
7	Menetapkan pengaturan upload file
8	Memanggil library upload
9	IF(file tidak berhasil di upload atau tidak sesuai pengaturan)
10	Mengirimkan pesan error
11	Menampilkan formulir permintaan darah
12	ELSE
13	Memasukkan data file ke dalam variabel data
14	END IF
15	Memasukkan data ke dalam database
16	Memperbarui data dari database
17	Menampilkan pesan berhasil ditambah
18	ELSE
19	Menampilkan validasi error
20	Menampilkan formulir permintaan darah
21	END IF
22	ELSE
23	Mengambil data dari database

24	Menampilkan formulir permintaan darah
25	END IF
26	Selesai

5.1.5 Perancangan Antarmuka

Pada bagian ini menjelaskan tentang contoh baku perancangan antarmuka yang akan diterapkan kedalam sistem. Perancangan antarmuka akan dijelaskan melalui gambar, sebagai berikut.

5.1.5.1 Halaman Menambah Alur Pengiriman Darah

Halaman menambah alur pengiriman darah merupakan halaman yang ditampilkan oleh sistem saat *administrator* ingin menambah data daftar alur pengiriman darah. Pada Gambar 5.6. merupakan rancangan antarmuka menambah alur pengiriman darah.



Gambar 5.6 Perancangan Antarmuka Menambah Alur Pengiriman Darah

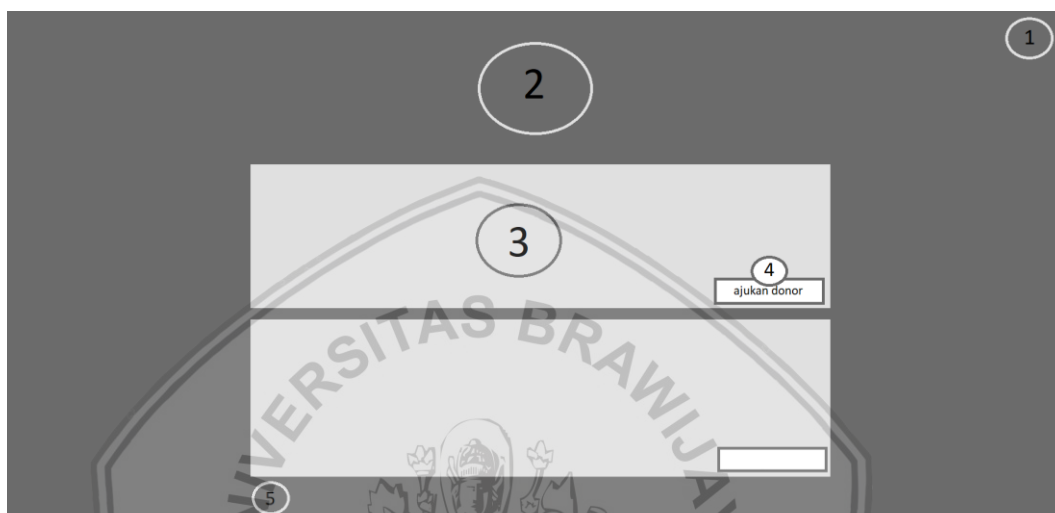
Pada gambar 5.6 terdapat beberapa bagian utama yang ditandai dengan angka yaitu:

1. Angka 1, bagian ini berisi nama aktor yang telah terotorisasi ke dalam sistem yang pada gambar tersebut adalah *administrator*, dan beserta *dropdown* yaitu menu *logout* untuk keluar dari sistem.
2. Angka 2, bagian ini merupakan simbol lonceng untuk melihat pemberitahuan permintaan darah baru.
3. Angka 3, bagian ini merupakan *sidebar* yang berisi menu-menu yang dapat diakses di dalam sistem.
4. Angka 4, bagian ini berisi formulir yang digunakan untuk menambah data alur pengiriman darah.
5. Angka 5, bagian ini berupa *button* yang berfungsi untuk menyimpan data setelah *administrator* menambah data.

- Angka 6, bagian ini berupa *button* yang berfungsi untuk kembali ke halaman sebelum *administrator* menambah data.

5.1.5.2 Halaman Melihat Informasi Butuh Darah

Halaman melihat informasi butuh darah merupakan halaman yang ditampilkan oleh sistem saat pendonor ingin melihat data darah yang sedang dibutuhkan sesuai dengan golongan darah pendonor. Pada Gambar 5.7 merupakan rancangan antarmuka melihat informasi butuh darah.



Gambar 5.7 Perancangan Antarmuka Melihat Informasi Butuh Darah

Pada gambar 5.7 terdapat beberapa bagian utama yang ditandai dengan angka yaitu:

- Angka 1, bagian ini merupakan *sidebar* yang berisi menu-menu yang dapat diakses di dalam sistem.
- Angka 2, bagian ini merupakan keterangan nama judul informasi.
- Angka 3, bagian ini berisi informasi data wali serta darah yang sedang dibutuhkan.
- Angka 4, bagian ini berupa *button* yang berfungsi untuk mengajukan donor untuk golongan darah tersebut.
- Angka 5, bagian ini berupa *pagination* untuk melihat informasi data darah yang dibutuhkan selanjutnya.

5.1.5.3 Halaman Menambah Permintaan Darah

Halaman menambah permintaan darah merupakan halaman yang ditampilkan oleh sistem saat petugas rumah sakit ingin mengajukan permintaan darah. Pada Gambar 5.8 merupakan rancangan antarmuka menambah permintaan darah.

The screenshot shows a web application interface for 'Data Permintaan' (Request Data). It features a sidebar on the left (labeled 2) and a main form area (labeled 3). The form contains the following fields: Nomor Permintaan, Kode Petugas, Nama Rumah Sakit, Nama Petugas, Alamat Rumah Sakit, Telepon Rumah Sakit, Nama Dokter, Jumlah Komponen, Jenis Komponen, Nama Pasien, Nomor Med. Rec, Umur, Agama, Jenis Kelamin, and Golongan Darah. At the bottom of the form are two buttons: 'submit' (labeled 4) and 'kembali' (labeled 5). A small number 1 is located in the top right corner of the browser window.

Gambar 5.8 Perancangan antarmuka menambah permohonan donor

Pada gambar 5.4 terdapat beberapa bagian utama yang ditandai dengan angka yaitu:

1. Angka 1, bagian ini berisi nama aktor yang telah terotorisasi ke dalam sistem yang pada gambar tersebut adalah *administrator*, dan beserta *dropdown* yaitu menu profil untuk melihat atau mengubah profil dan *logout* untuk keluar dari sistem.
2. Angka 2, bagian ini merupakan *sidebar* yang berisi menu-menu yang dapat diakses di dalam sistem.
3. Angka 3, bagian ini berisi formulir yang digunakan untuk menambah data permintaan darah.
4. Angka 4, bagian ini berupa *button* yang berfungsi untuk menyimpan data setelah petugas rumah sakit menambah data.
5. Angka 5, bagian ini berupa *button* yang berfungsi untuk kembali ke halaman sebelum petugas rumah sakit menambah data.

5.2 Implementasi

Bab implementasi membahas bagaimana implementasi sistem aplikasi manajemen Unit Transfusi Darah (UTD) berdasarkan metodologi dan perancangan. Implementasi yang akan dijabarkan adalah spesifikasi sistem, implementasi algoritme, dan implementasi antarmuka sistem.

5.2.1 Spesifikasi Sistem

Dalam mengimplementasikan perancangan untuk mewujudkan sistem ini, dibutuhkan perangkat keras dan perangkat lunak.

5.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan oleh sistem dijelaskan pada Tabel 6.1 dengan spesifikasi sebagai berikut :

Tabel 5.4 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Model</i>	ASUS FX553VD
<i>Processor</i>	Intel Core i7 7700HQ Processor
<i>Memory Size</i>	8GB with 2 x SO-DIMM socket, up to 32 GB SDRAM
<i>Graphics Processor</i>	NVIDIA GeForce GTX 1050, with 2GB / 4GB GDDR5 VRAM
<i>Hard Drive Size</i>	1 TB 5400 rpm

5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan oleh sistem dijelaskan pada Tabel 6.2 dengan spesifikasi sebagai berikut :

Tabel 5.5 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 Pro 64-bit
<i>Programming Language</i>	PHP, Javascript, CSS, HTML
<i>Programming Environment</i>	Sublime Text 3
<i>Framework</i>	Codeigniter
<i>Database Management System</i>	MySQL versi 5.0.12

5.2.2 Implementasi Algoritme

Pada implementasi algoritme ini, algoritme diubah menjadi kode program atau source code yang dapat dimengerti oleh komputer. Kode program dibuat menggunakan bahasa pemrograman PHP. Berikut ini merupakan beberapa sampel kode program.

5.2.2.1 Implementasi *Algoritme* Menambah Alur Pengiriman Darah

Tabel 5.1 merupakan implementasi algoritme untuk menambah alur pengiriman darah dari algoritme fungsi `add_alurpengiriman` pada kelas `Daftarpermintaanadm` dalam bentuk bahasa pemrograman php.

Tabel 5.6 Implementas *Algoritme* Menambah Alur Pengiriman Darah

1	public function add_alurpengiriman() {
2	if (\$this->input->post('submit', TRUE) ==
3	'Submit'){
4	\$this->form_validation->
5	set_rules('koderumahsakit', 'Nama Rumah Sakit',
6	'required');
7	\$this->form_validation->set_rules('status',
8	'Status', 'required');
9	\$this->form_validation->set_rules('petugas',
10	'Petugas', 'required');
11	
12	if (\$this->form_validation->run() == TRUE) {
13	\$alurpengiriman = array (
14	'idpermintaan' => \$this->uri->segment(3),
15	'koderumahsakit' => \$this->input->
16	post('koderumahsakit',TRUE),
17	'status' => \$this->input->
18	post('status',TRUE),
19	'petugas' => \$this->input->
20	post('petugas',TRUE)
21);
22	
23	\$statusminta = '';
24	if (\$alurpengiriman['status'] ==
25	'Permintaan disetujui') {
26	\$statusminta = 'DISETUJUI' ;
27	}else if (\$alurpengiriman['status'] ==
28	'Darah diterima rs') {
29	\$statusminta = 'DITERIMA' ;
30	}else if (\$alurpengiriman['status'] ==
31	'Permintaan selesai') {
32	\$statusminta = 'SELESAI' ;
33	}else if (\$alurpengiriman['status'] ==
34	'Permintaan dibatalkan' \$alurpengiriman['status']
35	== 'Permintaan dibatalkan dengan alasan') {
36	\$statusminta = 'DIBATALKAN' ;
37	}
38	
39	\$proses = \$this->m_admin->
40	insert('alurpengiriman', \$alurpengiriman);
41	\$this->m_admin->update('formpermintaan',
42	array('statusminta' => \$statusminta),
43	array('id_minta' =>
44	\$alurpengiriman['idpermintaan']));
45	

46	if (\$proses) {
47	echo "<script>alert('Data berhasil
48	ditambah!');</script>";
49	redirect('daftarpermintaanadm',
50	'refresh');
51	}
52	}
53	}
54	\$data['koderumahsakit'] = \$this->input-
55	>post('namarumahsakit',TRUE);
56	\$data['status'] = \$this->input-
57	>post('status',TRUE);
58	\$data['petugas'] = \$this->input-
59	>post('petugas',TRUE);
60	
61	\$data['header'] = "Formulir Menambah Alur
62	Pengiriman";
63	\$data['notifikasi'] = \$this->m_admin-
64	>get_notif();
65	\$this->template-
66	>admin('admin/form_alurpengiriman', \$data);
67	}

5.2.2.2 Implementasi *Algoritme* Melihat Informasi Butuh Darah

Tabel 5.1 merupakan implementasi algoritme untuk melihat informasi butuh darah dari algoritme fungsi butuh pada kelas Pendoror dalam bentuk bahasa pemrograman php.

Tabel 5.7 Implementas *Algoritme* Melihat Informasi Butuh Darah

1	public function butuh() {
2	\$form = \$this->uri->segment(3);
3	\$gol = \$this->session->userdata('goldar');
4	
5	\$pendonor = array();
6	if(\$gol == "O+"){
7	\$pendonor = ["O+", "A+", "B+", "AB+"];
8	}else if (\$gol == "A+"){
9	\$pendonor = ["A+", "AB+"];
10	}else if (\$gol == "B+"){
11	\$pendonor = ["B+", "AB+"];
12	}else if (\$gol == "AB+"){
13	\$pendonor = ["AB+"];
14	}else if (\$gol == "O-"){
15	\$pendonor = ["O+", "A+", "B+", "AB+", "O-", "A-
16	", "B-", "AB-"];
17	}else if (\$gol == "A-"){
18	\$pendonor = ["A+", "A-", "AB+", "AB-"];
19	}else if (\$gol == "B-"){
20	\$pendonor = ["B+", "B-", "AB+", "AB-"];
21	}else if (\$gol == "AB-"){
22	\$pendonor = ["AB+", "AB-"];
23	}
24	\$jumlahbaris = \$this->m_pendonor-
25	>baris_gol(\$pendonor);
26	
27	\$config['base_url'] =
28	'http://localhost/samutd/pendonor/butuh';
29	\$config['total_rows'] = \$jumlahbaris;
30	\$config['per_page'] = 2;
31	\$config['next_link'] = 'Selanjutnya';
32	\$config['prev_link'] = 'Sebelumnya';
33	
34	\$config['first_link'] = false;
35	\$config['last_link'] = false;
36	
37	\$data_gol = \$this->m_pendonor-
38	>get_data_donor(\$pendonor, \$config['per_page'],
39	\$form);
40	\$this->pagination->initialize(\$config);
41	
42	\$data['goldar1'] = \$data_gol;
43	\$this->load->view('pendonor/butuhview',
44	\$data);
45	}

5.2.2.3 Implementasi *Algoritme* Menambah Permintaan Darah

Tabel 5.1 merupakan implementasi algoritme untuk menambah permintaan darah dari algoritme fungsi `add_permintaan` pada kelas `Petugas` dalam bentuk bahasa pemrograman php.

Tabel 5.8 Implementas *Algoritme* Menambah Permintaan Darah

1	<code>public function add_permintaan(\$id) {</code>
2	
3	<code>if (\$this->input->post('submit', TRUE) ==</code>
4	<code>'Submit'){</code>
5	<code>\$this->form_validation-</code>
6	<code>>set_rules('namadr', 'Nama Dokter', 'required');</code>
7	<code>\$this->form_validation-</code>
8	<code>>set_rules('jmlh_kom', 'Jumlah Komponen',</code>
9	<code>'required');</code>
10	<code>\$this->form_validation-</code>
11	<code>>set_rules('jenkom', 'Jenis Komponen', 'required');</code>
12	<code>\$this->form_validation-</code>
13	<code>>set_rules('namapas', 'Nama Pasien', 'required');</code>
14	<code>\$this->form_validation-</code>
15	<code>>set_rules('nomed', 'Momor Med.Rec',</code>
16	<code>'required numeric');</code>
17	<code>\$this->form_validation-</code>
18	<code>>set_rules('umurpas', 'Umur Pasien', 'required');</code>
19	<code>\$this->form_validation-</code>
20	<code>>set_rules('agamapas', 'Agama Pasien', 'required');</code>
21	<code>\$this->form_validation->set_rules('jk',</code>
22	<code>'Jenis Kelamin', 'required');</code>
23	<code>\$this->form_validation-</code>
24	<code>>set_rules('gol_dar', 'Golongan Darah', 'required');</code>
25	<code>\$this->form_validation->set_rules('file',</code>
26	<code>'Upload Bukti Surat Dokter', 'required');</code>
27	
28	<code>if (\$this->form_validation->run() ==</code>
29	<code>TRUE) {</code>
30	<code>\$petugas = array (</code>
31	<code>'kode' => \$this->input-</code>
32	<code>>post('kode', TRUE),</code>
33	<code>'namarumahsakit' => \$this->input-</code>
34	<code>>post('namarumahsakit', TRUE),</code>
35	<code>'nmpj' => \$this->input-</code>
36	<code>>post('nmpj', TRUE),</code>
37	<code>'alamatrs' => \$this->input-</code>
38	<code>>post('alamatrs', TRUE),</code>
39	<code>'teleponrs' => \$this->input-</code>
40	<code>>post('teleponrs', TRUE));</code>

```

41
42             $formpermintaan = array (
43                 'idakun'          =>          $this->session-
44 >userdata('idpetugas'),
45                 'nomin'          =>          $this->input-
46 >post('nomin',TRUE),
47                 'namadr'         =>          $this->input-
48 >post('namadr',TRUE),
49                 'jmlh_kom'       =>          $this->input-
50 >post('jmlh_kom',TRUE),
51                 'jenkom'        =>          $this->input-
52 >post('jenkom',TRUE),
53                 'namapas'       =>          $this->input-
54 >post('namapas',TRUE),
55                 'nomed'         =>          $this->input-
56 >post('nomed',TRUE),
57                 'umurpas'       =>          $this->input-
58 >post('umurpas',TRUE),
59                 'agamapas'      =>          $this->input-
60 >post('agamapas',TRUE),
61                 'jk' => $this->input->post('jk',TRUE),
62                 'gol_dar'       =>          $this->input-
63 >post('gol_dar',TRUE),
64                 'file'         =>          =>
65 "SuratDokter"."_".trim(str_replace(" ", "",
66 date('dmY'))));
67
68                 $config['upload_path']          =
69 './uploads/';
70                 $config['allowed_types']         =
71 'pdf|doc|docx';
72                 $config['file_name']             =
73 $formpermintaan['file'];
74                 $this->load->library('upload',
75 $config);
76
77                 if ( ! $this->upload-
78 >do_upload('file')){
79                     $error = array('error' => $this-
80 >upload->display_errors());
81                     $this->templatepts-
82 >petugas('petugas/form_permintaan', $error);
83                 }else{
84                     $data = array('upload_data' =>
85 $this->upload->data());
86                 }

```

87	\$proses = \$this->m_petugas-
88	>insert('formpermintaan', \$formpermintaan);
89	\$this->m_petugas-
90	>update('akunpetugas', \$petugas, array('idpetugas' =>
91	\$id));
92	
93	if (\$proses) {
94	echo "<script>alert('Data berhasil
95	ditambah!');</script>";
96	}
97	redirect('petugas', 'refresh');
98	
99	}else{
100	\$data['kode'] = \$this->input-
101	>post('kode');
102	\$data['namarumahsakit'] = \$this-
103	>input->post('namarumahsakit');
104	\$data['nmpj'] = \$this->input-
105	>post('nmpj');
106	\$data['alamatrs'] = \$this->input-
107	>post('alamatrs');
108	\$data['teleponrs'] = \$this->input-
109	>post('teleponrs');
110	\$data['nomin'] = \$this->input-
111	>post('nomin');
112	\$data['namadr'] = \$this->input-
113	>post('namadr');
114	\$data['jmlh_kom'] = \$this->input-
115	>post('jmlh_kom');
116	\$data['jenkom'] = \$this->input-
117	>post('jenkom');
118	\$data['namapas'] = \$this->input-
119	>post('namapas');
120	\$data['nomed'] = \$this->input-
121	>post('nomed');
122	\$data['umurpas'] = \$this->input-
123	>post('umurpas');
124	\$data['agamapas'] = \$this->input-
125	>post('agamapas');
126	\$data['jk'] = \$this->input-
127	>post('jk');
128	\$data['gol_dar'] = \$this->input-
129	>post('gol_dar');
130	\$data['file'] = \$this->input-
131	>post('file');
132	

```

133         $this->templateptgs-
134 >petugas('petugas/form_permintaan', $data);
135     }
136     } else {
137         $key = $this->m_petugas->get_akun($id)-
138 >row();
139
140         $data['idakun'] = $key->idakun;
141         $data['kode'] = $key->kode;
142         $data['namarumahsakit'] = $key->
143 >namarumahsakit;
144         $data['nmpj'] = $key->nmpj;
145         $data['alamatrs'] = $key->alamatrs;
146         $data['teleponrs'] = $key->teleponrs;
147
148         $this->templateptgs-
149 >petugas('petugas/form_permintaan', $data);
150     }
151 }

```

5.2.3 Implementasi Antarmuka

Antarmuka sistem adalah sarana yang digunakan aktor untuk melakukan interaksi dengan sistem. Berikut penjelasan mengenai implementasi antarmuka pada Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD).

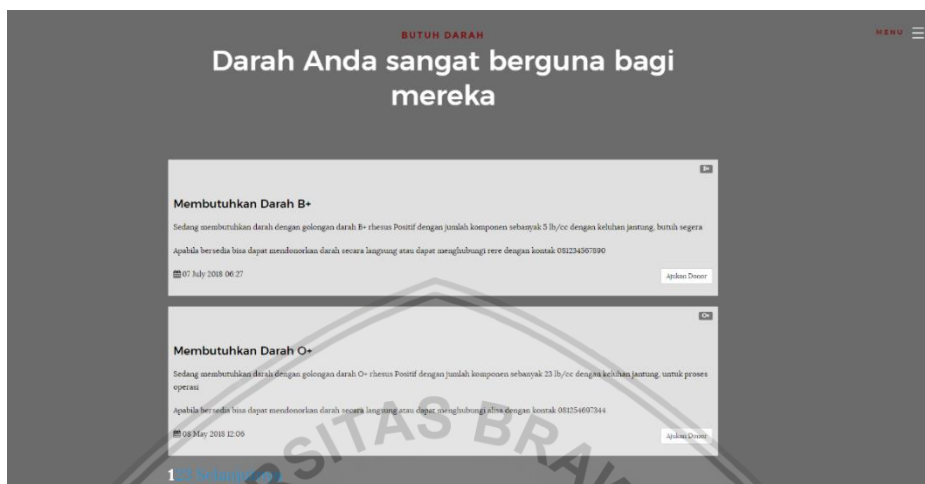
5.2.3.1 Implementasi Antarmuka Menambah Alur Pengiriman Darah

Antarmuka menambah alur pengiriman darah merupakan halaman yang ditampilkan oleh sistem saat *administrator* ingin menambah data alur pengiriman darah. Gambar 6.1 adalah implementasi antarmuka menambah alur pengiriman darah.

Gambar 5.9 Implementasi Antarmuka Menambah Alur Pengiriman Darah

5.2.3.2 Implementasi Antarmuka Melihat Informasi Butuh Darah

Antarmuka melihat informasi butuh darah merupakan halaman yang ditampilkan oleh sistem saat pendonor ingin melihat data butuh darah sesuai dengan golongan yang dapat didonorkan oleh pendonor. Gambar 6.2 adalah implementasi antarmuka melihat butuh darah.



Gambar 5.10 Implementasi Antarmuka Melihat Informasi Butuh Darah

5.2.3.3 Implementasi Antarmuka Menambah Permintaan Darah

Antarmuka menambah permintaan darah merupakan halaman yang ditampilkan oleh sistem saat petugas rumah sakit ingin menambahkan permintaan darah. Pada halaman ini terdapat formulir yang telah terisi untuk dicek kembali oleh petugas rumah sakit apakah data tersebut sudah benar dan lengkap atau belum dan terdapat beberapa yang harus diisi oleh petugas rumah sakit. Gambar 6.3 adalah implementasi antarmuka menambah permintaan darah.

Gambar 5.11 Implementasi Antarmuka Menambah Permintaan Darah

BAB 6 PENGUJIAN

Pada bab ini akan membahas pengujian dan analisis dari Sistem Aplikasi Manajemen Unit Transfusi Darah (UTD). Tujuan pengujian ini adalah untuk menemukan kesalahan yang terdapat pada sistem dan untuk memastikan bahwa sistem yang dibangun telah sesuai dengan yang dibutuhkan.

6.1 Pengujian Unit

Pengujian *white box* digunakan untuk melakukan pengecekan terhadap detail perancangan sistem, dimulai dengan struktur dari desain sistem secara prosedural kemudian dipecah menjadi beberapa kasus uji. Algoritme sistem disusun dalam bentuk *flow graph* untuk menentukan jumlah *cyclomatic complexity* (kompleksitas siklomatis) dan jalur independennya.

6.1.1 Pengujian Unit Menambah Alur Pengiriman Darah

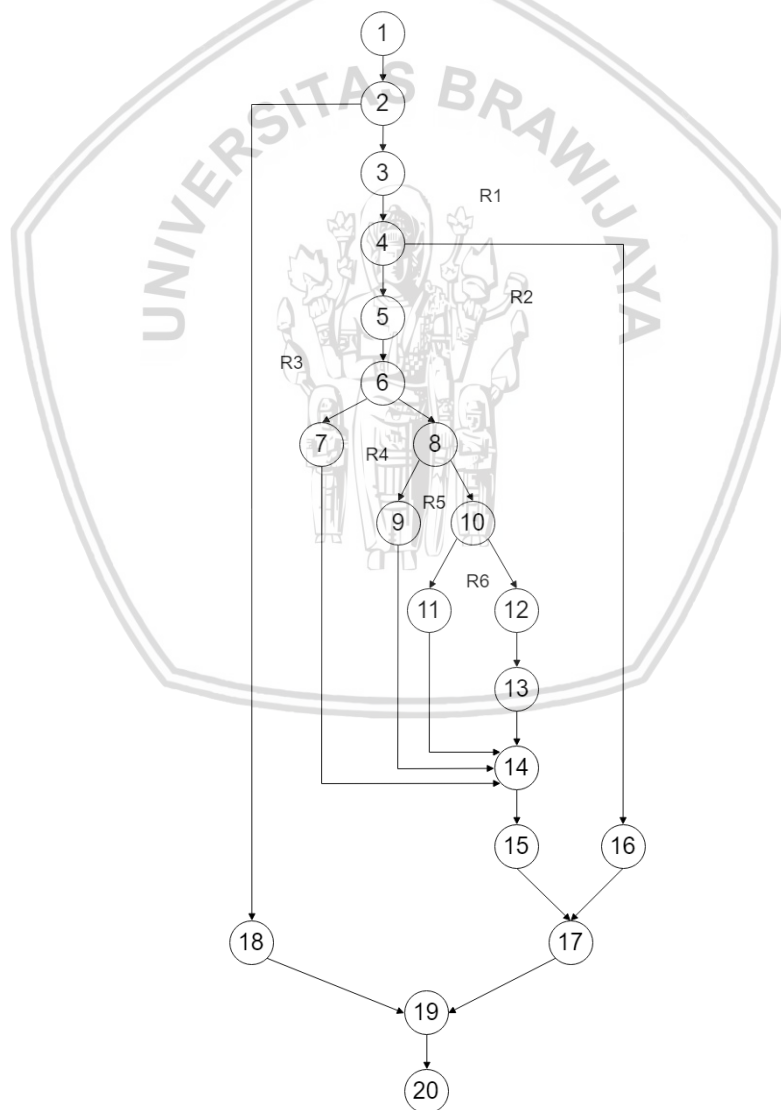
Tabel 6.1 merupakan pemaparan dari algoritme untuk melakukan proses menambah alur pengiriman darah dengan fungsi `add_alurpengiriman()` yang disertai dengan *node flowgraph*.

Tabel 6.1 Pembentukan Node Algoritme fungsi `add_alurpengiriman()`

Baris	Pseudocode	Node
1	Mulai	1
2	IF (nilai input dari submit = true)	2
3	Menetapkan peraturan untuk validasi form	3
4	IF (validasi = true)	4
5	Inisialisasi array variabel alur pengiriman dengan inputan dari form	5
6	IF (status alur pengiriman = permintaan disetujui)	6
7	Inisialisasi variabel status minta adalah disetujui	7
8	ELSE IF (status alur pengiriman = darah diterima rs)	8
9	Inisialisasi variabel status minta adalah diterima	9
10	ELSE IF (status alur pengiriman = permintaan selesai)	10
11	Inisialisasi variabel status minta adalah selesai	11
12	ELSE IF (status alur pengiriman = permintaan dibatalkan atau permintaan dibatalkan dengan alasan)	12
13	Inisialisasi variabel status minta adalah dibatalkan	13
14	END IF	14
15	Memasukkan data ke dalam database	15

16	Memperbarui data dari database	
17	Menampilkan pesan berhasil ditambah	
18	ELSE	16
19	Menampilkan validasi error	
20	END IF	17
21	ELSE	18
22	Menampilkan formulir menambah alur pengiriman	
23	END IF	19
24	Selesai	20

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6.1, maka diperoleh bentuk *flowgraph* seperti dalam Gambar 6.1.



Gambar 6.1 Flowgraph Algoritme fungsi add_alurpengiriman()

Berdasarkan jumlah *flowgraph* yang telah diperoleh, maka ditentukan enam jalur independen dan jumlah kompleksitas siklomatis, seperti berikut:

Jalur 1: 1-2-18-19-20

Jalur 2: 1-2-3-4-16-17-20

Jalur 3: 1-2-3-4-5-6-7-14-15-17-19-20

Jalur 4: 1-2-3-4-5-6-8-9-14-15-17-19-20

Jalur 5: 1-2-3-4-5-6-8-10-11-14-15-17-19-20

Jalur 6: 1-2-3-4-5-6-8-10-12-13-14-15-17-19-20

Perhitungan cyclomatic complexity:

$V(G) = \text{Jumlah Region} = 6$

$V(G) = E - N + 2 = 24 - 20 + 2 = 6$

$V(G) = P + 1 = 5 + 1 = 6$

Berdasarkan jalur yang telah didefinisikan, maka dibentuklah kasus ujinya. Tabel 6.2 merupakan pemaparan kasus uji dari algoritme fungsi `add_alurpengiriman()`.

Tabel 6.2 Kasus Uji Menambah Alur Pengiriman Darah

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Nilai input dari submit = false	Menampilkan formulir menambah alur pengiriman.	Menampilkan formulir menambah alur pengiriman.	Valid
2	Nilai input dari submit = true dan validasi = false	Menetapkan peraturan validasi formulir dan menampilkan validasi error	Menetapkan peraturan validasi formulir dan menampilkan validasi error	Valid
3	Nilai input dari submit = true, validasi = true, dan status alur pengiriman = permintaan disetujui	Menginisialisasi variabel status minta adalah disetujui dan menampilkan pesan berhasil ditambah	Menginisialisasi variabel status minta adalah disetujui dan menampilkan pesan berhasil ditambah	Valid
4	Nilai input dari submit = true, validasi = true, dan status alur	Menginisialisasi variabel status minta adalah diterima dan	Menginisialisasi variabel status minta adalah diterima dan	Valid

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
	pengiriman = darah diterima rs	menampilkan pesan berhasil ditambah	menampilkan pesan berhasil ditambah	
5	Nilai input dari submit = true, validasi = true, dan status alur pengiriman = permintaan selesai	Menginisialisasi variabel status minta adalah selesai dan menampilkan pesan berhasil ditambah	Menginisialisasi variabel status minta adalah selesai dan menampilkan pesan berhasil ditambah	Valid
6	Nilai input dari submit = true, validasi = true, dan status alur pengiriman = permintaan dibatalkan atau permintaan dibatalkan dengan alasan	Menginisialisasi variabel status minta adalah dibatalkan dan menampilkan pesan berhasil ditambah	Menginisialisasi variabel status minta adalah dibatalkan dan menampilkan pesan berhasil ditambah	Valid

6.1.2 Pengujian Unit Melihat Informasi Butuh Darah

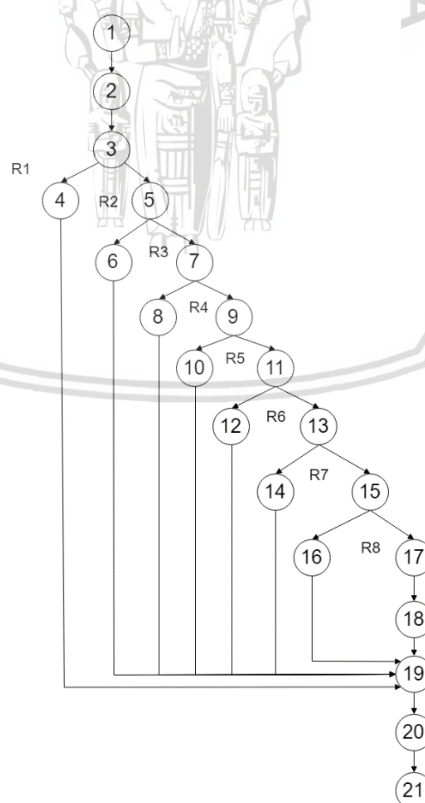
Tabel 6.3 merupakan pemaparan dari algoritme butuh() yang disertai dengan *node flowgraph*.

Tabel 6.3 Perancangan Algoritme Melihat Informasi Butuh Darah

Baris	Pseudocode	Node
1	Mulai	1
2	Mengambil session golongan darah dari array	2
3	Inisialisasi variabel pendonor berupa array	
4	IF (golongan darah = O+)	3
5	Inisialisasi variabel pendonor berupa O+, A+, B+, AB+	4
6	ELSE IF (golongan darah = A+)	5
7	Inisialisasi variabel pendonor berupa A+, AB+	6
8	ELSE IF (golongan darah = B+)	7
9	Inisialisasi variabel pendonor berupa B+, AB+	8
10	ELSE IF (golongan darah = AB+)	9
11	Inisialisasi variabel pendonor berupa AB+	10
12	ELSE IF (golongan darah = O-)	11

13	Inisialisasi variabel pendonor berupa O+, A+, B+, AB+, O-, A-, B-, AB-	12
14	ELSE IF (golongan darah = A-)	13
15	Inisialisasi variabel pendonor berupa A+, AB+, A-, AB-	14
16	ELSE IF (golongan darah = B-)	15
17	Inisialisasi variabel pendonor berupa B+, AB+, B-, AB-	16
18	ELSE IF (golongan darah = AB-)	17
19	Inisialisasi variabel pendonor berupa AB+, AB-	18
20	END IF	19
21	Mengambil jumlah baris data melalui model	20
22	Konfigurasi untuk settingan pagination	
23	Mengambil data golongan darah sesuai kondisi dari database	
24	Menampilkan data sesuai offset dan aturan per halaman yang ditentukan	
25	Selesai	21

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6.3, maka diperoleh bentuk *flowgraph* seperti dalam Gambar 6.2.



Gambar 6.2 Flowgraph Algoritme fungsi butuh()

Berdasarkan jumlah *flowgraph* yang telah diperoleh, maka ditentukan delapan jalur independen dan jumlah kompleksitas siklomatis, seperti berikut:

Jalur 1: 1-2-3-4-19-20-21

Jalur 2: 1-2-3-5-6-19-20-21

Jalur 3: 1-2-3-5-7-8-19-20-21

Jalur 4: 1-2-3-5-7-9-10-19-20-21

Jalur 5: 1-2-3-5-7-9-11-12-19-20-21

Jalur 6: 1-2-3-5-7-9-11-13-14-19-20-21

Jalur 7: 1-2-3-5-7-9-11-13-15-16-19-20-21

Jalur 8: 1-2-3-5-7-9-11-13-15-17-18-19-20-21

Perhitungan cyclomatic complexity:

$$V(G) = \text{Jumlah Region} = 8$$

$$V(G) = E - N + 2 = 27 - 21 + 2 = 8$$

$$V(G) = P + 1 = 7 + 1 = 8$$

Berdasarkan jalur yang telah didefinisikan, maka dibentuklah kasus ujinya. Tabel 6.4 merupakan pemaparan kasus uji dari algoritme fungsi butuh().

Tabel 6.4 Kasus Uji Melihat Informasi Butuh Darah

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Mengambil session golongan darah dengan nilai input dari golongan darah = O+	Menginisialisasi variabel pendonor berupa O+, A+, B+, AB+, mengambil data dan menampilkan data sesuai offset dan aturan	Menginisialisasi variabel pendonor berupa O+, A+, B+, AB+, mengambil data dan menampilkan data sesuai offset dan aturan	Valid
2	Mengambil session golongan darah dengan nilai input dari golongan darah = A+	Menginisialisasi variabel pendonor berupa A+, AB+, mengambil data dan menampilkan data sesuai offset dan aturan	Menginisialisasi variabel pendonor berupa A+, AB+, mengambil data dan menampilkan data sesuai offset dan aturan	Valid
3	Mengambil session golongan darah dengan nilai input dari	Menginisialisasi variabel pendonor berupa B+, AB+, mengambil data dan	Menginisialisasi variabel pendonor berupa B+, AB+, mengambil data	Valid

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
	golongan darah = B+	menampilkan data sesuai offset dan aturan	dan menampilkan data sesuai offset dan aturan	
4	Mengambil session golongan darah dengan nilai input dari golongan darah = AB+	Menginisialisasi variabel pendonor berupa AB+, mengambil data dan menampilkan data sesuai offset dan aturan	Menginisialisasi variabel pendonor berupa AB+, mengambil data dan menampilkan data sesuai offset dan aturan	Valid
5	Mengambil session golongan darah dengan nilai input dari golongan darah = O-	Menginisialisasi variabel pendonor berupa O+, A+, B+, AB+, O-, A-, B-, AB-, mengambil data dan menampilkan data sesuai offset dan aturan	Menginisialisasi variabel pendonor berupa O+, A+, B+, AB+, O-, A-, B-, AB-, mengambil data dan menampilkan data sesuai offset dan aturan	Valid
6	Mengambil session golongan darah dengan nilai input dari golongan darah = A-	Menginisialisasi variabel pendonor berupa A+, AB+, A-, AB-, mengambil data dan menampilkan data sesuai offset dan aturan	Menginisialisasi variabel pendonor berupa A+, AB+, A-, AB-, mengambil data dan menampilkan data sesuai offset dan aturan	Valid
7	Mengambil session golongan darah dengan nilai input dari golongan darah = B-	Menginisialisasi variabel pendonor berupa B+, AB+, B-, AB-, mengambil data dan menampilkan data sesuai offset dan aturan	Menginisialisasi variabel pendonor berupa B+, AB+, B-, AB-, mengambil data dan menampilkan data sesuai offset dan aturan	Valid
8	Mengambil session golongan darah dengan nilai input dari golongan darah = AB-	Menginisialisasi variabel pendonor berupa AB+, AB-, mengambil data dan menampilkan data	Menginisialisasi variabel pendonor berupa AB+, AB-, AB-, mengambil data dan menampilkan data	Valid

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
		sesuai offset dan aturan	data sesuai offset dan aturan	

6.1.3 Pengujian Unit Menambah Permintaan Darah

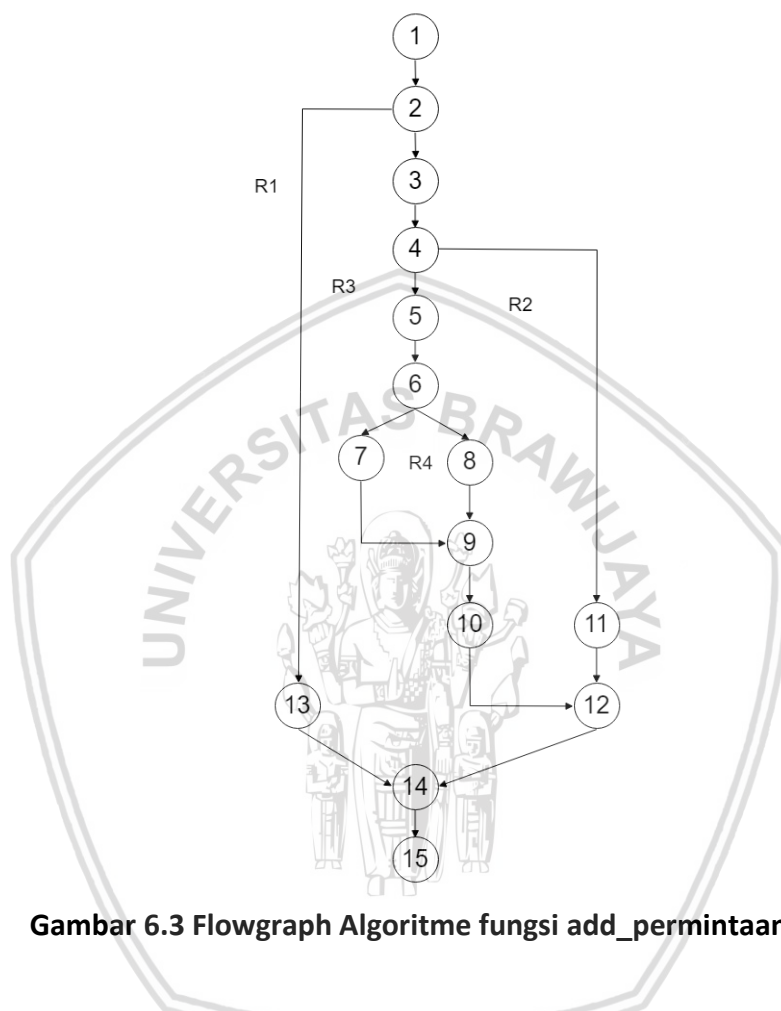
Tabel 6.5 merupakan pemaparan dari algoritme `add_permintaan()` yang disertai dengan *node flowgraph*.

Tabel 6.5 Perancangan Algoritme Menambah Permintaan Darah

Baris	Pseudocode	Node
1	Mulai	1
2	IF (nilai input dari submit = true)	2
3	Menetapkan peraturan untuk validasi form	3
4	IF (validasi = true)	4
5	Inisialisasi array variabel petugas dengan inputan dari form	5
6	Inisialisasi array variabel formpermintaan dengan inputan dari form	
7	Menetapkan pengaturan upload file	
8	Memanggil library upload	
9	IF(file tidak berhasil di upload atau tidak sesuai pengaturan)	6
10	Mengirimkan pesan error	7
11	Menampilkan formulir permintaan darah	
12	ELSE	8
13	Memasukkan data file ke dalam variabel data	
14	END IF	9
15	Memasukkan data ke dalam database	10
16	Memperbarui data dari database	
17	Menampilkan pesan berhasil ditambah	
18	ELSE	11
19	Menampilkan validasi error	
20	Menampilkan formulir permintaan darah	
21	END IF	12
22	ELSE	13
23	Mengambil data dari database	
24	Menampilkan formulir permintaan darah	

25	END IF	14
26	Selesai	15

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6., maka diperoleh bentuk *flowgraph* seperti dalam Gambar 6.2.



Gambar 6.3 Flowgraph Algoritme fungsi add_permintaan()

Berdasarkan jumlah *flowgraph* yang telah diperoleh, maka ditentukan delapan jalur independen dan jumlah kompleksitas siklomatis, seperti berikut:

Jalur 1: 1-2-13-14-15

Jalur 2: 1-2-3-4-11-12-14-15

Jalur 3: 1-2-3-4-5-6-7-9-10-12-14-15

Jalur 4: 1-2-3-4-5-6-8-9-10-12-14-15

Perhitungan cyclomatic complexity:

$$V(G) = \text{Jumlah Region} = 4$$

$$V(G) = E - N + 2 = 17 - 15 + 2 = 4$$

$$V(G) = P + 1 = 3 + 1 = 4$$

Berdasarkan jalur yang telah didefinisikan, maka dibentuklah kasus ujinya. Tabel 6.6 merupakan pemaparan kasus uji dari algoritme fungsi `add_permintaan()`.

Tabel 6.6 Kasus Uji Menambah Permintaan Darah

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Nilai input dari submit = false	Mengambil data dari database dan menampilkan formulir permintaan darah	Mengambil data dari database dan menampilkan formulir permintaan darah	Valid
2	Nilai input dari submit = true dan validasi = false	Menampilkan validasi error dan menampilkan formulir permintaan darah	Menampilkan validasi error dan menampilkan formulir permintaan darah	Valid
3	Nilai input dari submit = true, validasi = true, dan file upload tidak berhasil diupload	Mengirimkan pesan error dan menampilkan formulir permintaan darah	Mengirimkan pesan error dan menampilkan formulir permintaan darah	Valid
4	Nilai input dari submit = true, validasi = true, dan file upload berhasil diupload	Memasukkan data ke dalam variabel data dan database serta menampilkan pesan data berhasil ditambah	Memasukkan data ke dalam variabel data dan database serta menampilkan pesan data berhasil ditambah	Valid

6.2 Pengujian Validasi

Pengujian validasi akan dilakukan pada semua kebutuhan sistem dengan menggunakan metode *black box testing*. Pengujian validasi dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik dan tidak ada error yang terjadi. Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan seluruh kebutuhan yang telah ditetapkan atau tidak. Proses pengujian validasi mengacu pada daftar kebutuhan fungsional yang telah ditetapkan berdasarkan hasil proses analisis kebutuhan.

Tabel 6.7 Kasus Uji Pendaftaran

Nama Kasus Uji	Kasus Uji Pendaftaran
Kode Kebutuhan	SAM-F-01
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Daftar”. 2. Aktor memasukkan kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> pada formulir. 3. Aktor memilih tombol “Daftar”.
Hasil yang diharapkan	Sistem akan mendaftarkan aktor ke dalam sistem dan menyimpan data kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> .
Hasil	Sistem mendaftarkan aktor ke dalam sistem dan menyimpan data kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> .
Status	Valid

Tabel 6.8 Kasus Uji Pendaftaran

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Pendaftaran
Kode Kebutuhan	SAM-F-01
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Daftar”. 2. Aktor memasukkan kode pendonor, nomor identitas, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> pada formulir. 3. Aktor memilih tombol “Daftar”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.9 Kasus Uji Pendaftaran

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Pendaftaran
Kode Kebutuhan	SAM-F-01
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Daftar”. 2. Aktor memasukkan kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> pada formulir. 3. Aktor memilih tombol “Daftar”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.10 Kasus Uji Login

Nama Kasus Uji	Kasus Uji Login
Kode Kebutuhan	SAM-F-02
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “login”. 2. Aktor memasukkan <i>username</i> = rara dan <i>password</i> = 25163703 pada formulir login. 3. Aktor memilih tombol “Login”.
Hasil yang diharapkan	Aktor akan terautorisasi dan dapat mengakses sistem sebagai <i>administrator</i> , pendonor, atau petugas rumah sakit.
Hasil	Aktor terautorisasi dan dapat mengakses sistem sebagai <i>administrator</i> , pendonor, atau petugas rumah sakit.
Status	Valid

Tabel 6.11 Kasus Uji Login

Alternatif 1: *Username* dan *password* tidak sesuai

Nama Kasus Uji	Kasus Uji Login
Kode Kebutuhan	SAM-F-02
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “login”. 2. Aktor memasukkan <i>username</i> = rara dan <i>password</i> = haha pada formulir login. 3. Aktor memilih tombol “Login”.

Hasil yang diharapkan	Sistem akan menampilkan pesan peringatan " <i>username dan password tidak valid</i> ".
Hasil	Sistem menampilkan pesan peringatan " <i>username dan password tidak valid</i> ".
Status	Valid

Tabel 6.12 Kasus Uji Login

Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Login
Kode Kebutuhan	SAM-F-02
Prosedur	1. Aktor memilih tombol " <i>login</i> ". 2. Aktor memasukkan <i>username</i> = rara pada formulir <i>login</i> . 3. Aktor memilih tombol " <i>Login</i> ".
Hasil yang diharapkan	Sistem akan menampilkan pesan peringatan " <i>please fill out this field</i> ".
Hasil	Sistem menampilkan pesan peringatan " <i>please fill out this field</i> ".
Status	Valid

Tabel 6.13 Kasus Uji Logout

Nama Kasus Uji	Kasus Uji Logout
Kode Kebutuhan	SAM-F-03
Prosedur	1. Aktor terotorisasi ke dalam sistem 2. Aktor memilih menu logout
Hasil yang diharapkan	Sistem akan menghapus <i>session</i> .
Hasil	Sistem menghapus <i>session</i> .
Status	Valid

Tabel 6.14 Kasus Uji Melihat Stok Darah

Nama Kasus Uji	Kasus Uji Melihat Stok Darah
Kode Kebutuhan	SAM-F-04
Prosedur	1. Aktor memilih menu beranda
Hasil yang diharapkan	Sistem akan menampilkan tabel stok darah yang berisi nama produk, keterangan produk, dan jumlah darah pada semua golongan darah.
Hasil	Sistem menampilkan tabel stok darah yang berisi nama produk, keterangan produk, dan jumlah darah pada semua golongan darah.
Status	Valid

Tabel 6.15 Kasus Uji Melihat Alur Donor

Nama Kasus Uji	Kasus Uji Melihat Alur Donor
Kode Kebutuhan	SAM-F-05
Prosedur	1. Aktor memilih menu informasi
Hasil yang diharapkan	Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan poin-poin yang menjelaskan alur donor.
Hasil	Sistem menampilkan halaman informasi. Halaman tersebut menampilkan poin-poin yang menjelaskan alur donor.
Status	Valid

Tabel 6.16 Kasus Uji Melihat Informasi Setelah Donor

Nama Kasus Uji	Kasus Uji Melihat Informasi Setelah Donor
Kode Kebutuhan	SAM-F-06
Prosedur	1. Aktor memilih menu informasi
Hasil yang diharapkan	Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan poin-poin yang menjelaskan apa saja yang harus dilakukan setelah melakukan donor darah.
Hasil	Sistem menampilkan halaman informasi. Halaman tersebut menampilkan poin-poin yang menjelaskan apa saja yang harus dilakukan setelah melakukan donor darah.
Status	Valid

Tabel 6.17 Kasus Uji Menambah Pesan Hubungi

Nama Kasus Uji	Kasus Uji Menambah Pesan Hubungi
Kode Kebutuhan	SAM-F-07
Prosedur	1. Aktor memilih menu hubungi kami. 2. Aktor memasukkan nama lengkap, email, nomor telepon dan pesan yang akan dikirimkan pada formulir. 3. Aktor memilih tombol "submit".
Hasil yang diharapkan	Sistem akan menyimpan data nama lengkap, email, nomor telepon dan pesan.
Hasil	Sistem menyimpan data nama lengkap, email, nomor telepon dan pesan.
Status	Valid

Tabel 6.18 Kasus Uji Menambah Pesan Hubungi

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Pesan Hubungi
Kode Kebutuhan	SAM-F-07
Prosedur	1. Aktor memilih menu hubungi kami. 2. Aktor memasukkan nama lengkap, nomor telepon dan pesan yang akan dikirimkan pada formulir. 3. Aktor memilih tombol “submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan peringatan “ <i>please fill out this field</i> ”.
Hasil	Sistem menampilkan pesan peringatan “ <i>please fill out this field</i> ”.
Status	Valid

Tabel 6.19 Kasus Uji Melihat Daftar Pengguna

Nama Kasus Uji	Kasus Uji Melihat Daftar Pengguna
Kode Kebutuhan	SAM-F-08
Prosedur	1. Aktor memilih menu informasi
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar pengguna berupa tabel dengan data berupa <i>username</i> dan level pengguna.
Hasil	Sistem menampilkan halaman daftar pengguna berupa tabel dengan data berupa <i>username</i> dan level pengguna.
Status	Valid

Tabel 6.20 Kasus Uji Menambah Stok Darah

Nama Kasus Uji	Kasus Uji Menambah Stok Darah
Kode Kebutuhan	SAM-F-09
Prosedur	1. Aktor memilih tombol “tambah stok darah”. 2. Aktor mengisi formulir yang berisi produk, golongan darah, jumlah stok darah. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan data produk, golongan darah, jumlah stok darah.
Hasil	Sistem menyimpan data produk, golongan darah, jumlah stok darah.
Status	Valid

Tabel 6.21 Kasus Uji Menghapus Permintaan Darah

Alternatif 1: Aktor memilih tombol “cancel”

Nama Kasus Uji	Kasus Uji Menghapus Permintaan Darah
Kode Kebutuhan	SAM-F-09
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah stok darah”. 2. Aktor mengisi formulir yang berisi produk, golongan darah, jumlah stok darah. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan Permintaan Darah dan menampilkan halaman daftar Permintaan Darah.
Hasil	Sistem membatalkan proses penghapusan Permintaan Darah dan menampilkan halaman daftar Permintaan Darah.
Status	Valid

Tabel 6.22 Kasus Uji Menambah Pengguna

Nama Kasus Uji	Kasus Uji Menambah Pengguna
Kode Kebutuhan	SAM-F-10
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengguna”. 2. Aktor memasukkan kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> pada formulir pengguna. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.23 Kasus Uji Menambah Pengguna

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Pengguna
Kode Kebutuhan	SAM-F-10
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengguna”. 2. Aktor memasukkan kode pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> pada formulir pengguna.

	3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.24 Kasus Uji Menambah Pengguna

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Pengguna
Kode Kebutuhan	SAM-F-10
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol "tambah pengguna". 2. Aktor memasukkan kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, email, <i>username</i> dan <i>password</i> pada formulir pengguna. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.25 Kasus Uji Mengubah Pengguna

Nama Kasus Uji	Kasus Uji Mengubah Pengguna
Kode Kebutuhan	SAM-F-11
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol "tambah pengguna". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan data pengguna dan menampilkan halaman daftar pengguna.
Hasil	Sistem menyimpan data pengguna dan menampilkan halaman daftar pengguna.
Status	Valid

Tabel 6.26 Kasus Uji Mengubah Pengguna**Alternatif 1: Salah satu data kosong**

Nama Kasus Uji	Kasus Uji Mengubah Pengguna
Kode Kebutuhan	SAM-F-11
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengguna”. 2. Aktor mengubah data sesuai yang diinginkan 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.27 Kasus Uji Mengubah Pengguna**Alternatif 2: Salah satu data belum terisi dengan benar**

Nama Kasus Uji	Kasus Uji Mengubah Pengguna
Kode Kebutuhan	SAM-F-11
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengguna”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.28 Kasus Uji Menghapus Pengguna

Nama Kasus Uji	Kasus Uji Menghapus Pengguna
Kode Kebutuhan	SAM-F-12
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menghapus pengguna dan menampilkan halaman daftar pengguna.
Hasil	Sistem menghapus pengguna dan menampilkan halaman daftar pengguna.

Status	Valid
--------	-------

Tabel 6.29 Kasus Uji Menghapus Pengguna

Alternatif 1: Aktor memilih tombol “cancel”

Nama Kasus Uji	Kasus Uji Menghapus Pengguna
Kode Kebutuhan	SAM-F-12
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan pengguna dan menampilkan halaman daftar pengguna.
Hasil	Sistem membatalkan proses penghapusan pengguna dan menampilkan halaman daftar pengguna.
Status	Valid

Tabel 6.30 Kasus Uji Melihat Daftar Permintaan Darah

Nama Kasus Uji	Kasus Uji Melihat Daftar Permintaan Darah
Kode Kebutuhan	SAM-F-13
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu petugas 2. Aktor memilih sub menu permintaan darah
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar permintaan darah berupa tabel dengan data berupa tanggal, status, nama rumah sakit, nomor permintaan, golongan darah, jumlah komponen, dan jenis komponen.
Hasil	Sistem menampilkan halaman daftar permintaan darah berupa tabel dengan data berupa tanggal, status, nama rumah sakit, nomor permintaan, golongan darah, jumlah komponen, dan jenis komponen.
Status	Valid

Tabel 6.31 Kasus Uji Menambah Permintaan Darah

Nama Kasus Uji	Kasus Uji Menambah Permintaan Darah
Kode Kebutuhan	SAM-F-14
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah permintaan”.

	<p>2. Aktor memasukkan nomor permintaan, nama rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter pada formulir permintaan.</p> <p>3. Aktor memilih tombol "Submit".</p>
Hasil yang diharapkan	Sistem akan menyimpan nomor permintaan, nama rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter.
Hasil	Sistem menyimpan nomor permintaan, nama rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter.
Status	Valid

Tabel 6.32 Kasus Uji Menambah Permintaan Darah

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Permintaan Darah
Kode Kebutuhan	SAM-F-14
Prosedur	<p>1. Aktor memilih tombol "tambah permintaan".</p> <p>2. Aktor memasukkan nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter pada formulir permintaan.</p> <p>3. Aktor memilih tombol "Submit".</p>
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.33 Kasus Uji Menambah Permintaan Darah

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Permintaan Darah
Kode Kebutuhan	SAM-F-14
Prosedur	1. Aktor memilih tombol "tambah permintaan".

	2. Aktor memasukkan nomor permintaan, nama rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter pada formulir permintaan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.34 Kasus Uji Menambah Permintaan Darah

Alternatif 3: Upload file gagal atau tidak sesuai dengan peraturan

Nama Kasus Uji	Kasus Uji Menambah Permintaan Darah
Kode Kebutuhan	SAM-F-14
Prosedur	1. Aktor memilih tombol "tambah permintaan". 2. Aktor memasukkan nomor permintaan, nama rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, dan file bukti surat dokter pada formulir permintaan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan error.
Hasil	Sistem menampilkan pesan error.
Status	Valid

Tabel 6.35 Kasus Uji Menambah Alur Pengiriman Darah

Nama Kasus Uji	Kasus Uji Menambah Alur Pengiriman Darah
Kode Kebutuhan	SAM-F-15
Prosedur	1. Aktor memilih tombol "tambah". 2. Aktor memasukkan nama rumah sakit, status, dan petugas pada formulir alur pengiriman darah. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan nama rumah sakit, status, dan petugas dan menampilkan halaman daftar permintaan darah.

Hasil	Sistem menyimpan nama rumah sakit, status, dan petugas dan menampilkan halaman daftar permintaan darah.
Status	Valid

Tabel 6.36 Kasus Uji Menambah Menambah Alur Pengiriman Darah

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Alur Pengiriman Darah
Kode Kebutuhan	SAM-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah”. 2. Aktor memasukkan status, dan petugas pada formulir alur pengiriman darah. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan nama rumah sakit, status, dan petugas dan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menyimpan nama rumah sakit, status, dan petugas dan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.37 Kasus Uji Menambah Menambah Alur Pengiriman Darah

Alternatif 2: Status alur pengiriman bernilai permintaan disetujui

Nama Kasus Uji	Kasus Uji Menambah Alur Pengiriman Darah
Kode Kebutuhan	SAM-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah”. 2. Aktor memasukkan nama rumah sakit, status, dan petugas pada formulir alur pengiriman darah. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status disetujui pada halaman daftar permintaan darah.
Hasil	Sistem menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status disetujui pada halaman daftar permintaan darah.
Status	Valid

Tabel 6.38 Kasus Uji Menambah Menambah Alur Pengiriman Darah**Alternatif 3: Status alur pengiriman bernilai darah diterima rs**

Nama Kasus Uji	Kasus Uji Menambah Alur Pengiriman Darah
Kode Kebutuhan	SAM-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah”. 2. Aktor memasukkan nama rumah sakit, status, dan petugas pada formulir alur pengiriman darah. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status diterima pada halaman daftar permintaan darah.
Hasil	Sistem menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status diterima pada halaman daftar permintaan darah.
Status	Valid

Tabel 6.39 Kasus Uji Menambah Menambah Alur Pengiriman Darah**Alternatif 4: Status alur pengiriman bernilai permintaan selesai**

Nama Kasus Uji	Kasus Uji Menambah Alur Pengiriman Darah
Kode Kebutuhan	SAM-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah”. 2. Aktor memasukkan nama rumah sakit, status, dan petugas pada formulir alur pengiriman darah. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status selesai pada halaman daftar permintaan darah.
Hasil	Sistem menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status selesai pada halaman daftar permintaan darah.
Status	Valid

Tabel 6.40 Kasus Uji Menambah Menambah Alur Pengiriman Darah**Alternatif 5: Status alur pengiriman bernilai permintaan dibatalkan atau permintaan dibatalkan dengan alasan**

Nama Kasus Uji	Kasus Uji Menambah Alur Pengiriman Darah
Kode Kebutuhan	SAM-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah”. 2. Aktor memasukkan nama rumah sakit, status, dan petugas pada formulir alur pengiriman darah.

	3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status dibatalkan pada halaman daftar permintaan darah.
Hasil	Sistem menyimpan nama rumah sakit, status, dan petugas dan menampilkan status bernilai status dibatalkan pada halaman daftar permintaan darah.
Status	Valid

Tabel 6.41 Kasus Uji Melihat Detail Permintaan Darah

Nama Kasus Uji	Kasus Uji Melihat Detail Permintaan Darah
Kode Kebutuhan	SAM-F-16
Prosedur	1. Aktor memilih tombol "lihat".
Hasil yang diharapkan	Sistem akan menampilkan halaman detail permintaan darah dengan data berupa waktu permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur pasien, agama pasien, jenis kelamin, dan file bukti surat.
Hasil	Sistem menampilkan halaman detail permintaan darah dengan data berupa waktu permintaan, nomor permintaan, status permintaan, nama rumah sakit, alamat rumah sakit, telepon rumah sakit, nama dokter, golongan darah, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur pasien, agama pasien, jenis kelamin, dan file bukti surat.
Status	Valid

Tabel 6.42 Kasus Uji Mengubah Data Permintaan Darah

Nama Kasus Uji	Kasus Uji Mengubah Data Permintaan Darah
Kode Kebutuhan	SAM-F-17
Prosedur	1. Aktor memilih tombol "edit". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan data permintaan darah dan menampilkan halaman daftar permintaan darah.

Hasil	Sistem menyimpan data permintaan darah dan menampilkan halaman daftar permintaan darah.
Status	Valid

Tabel 6.43 Kasus Uji Mengubah Data Permintaan Darah

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Mengubah Data Permintaan Darah
Kode Kebutuhan	SAM-F-17
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.44 Kasus Uji Mengubah Data Permintaan Darah

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Mengubah Data Permintaan Darah
Kode Kebutuhan	SAM-F-17
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.45 Kasus Uji Melihat Daftar Data Petugas

Nama Kasus Uji	Kasus Uji Melihat Daftar Data Petugas
Kode Kebutuhan	SAM-F-18
Prosedur	1. Aktor memilih menu petugas 2. Aktor memilih sub menu Data Petugas
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar data petugas berupa tabel dengan data berupa nama

	rumah sakit, nama petugas, alamat rumah sakit, dan telepon rumah sakit.
Hasil	Sistem menampilkan halaman daftar data petugas berupa tabel dengan data berupa nama rumah sakit, nama petugas, alamat rumah sakit, dan telepon rumah sakit.
Status	Valid

Tabel 6.46 Kasus Uji Mengubah Data Data Petugas

Nama Kasus Uji	Kasus Uji Mengubah Data Data Petugas
Kode Kebutuhan	SAM-F-19
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “ <i>Submit</i> ”.
Hasil yang diharapkan	Sistem akan menyimpan data petugas dan menampilkan halaman daftar data petugas.
Hasil	Sistem menyimpan data petugas dan menampilkan halaman daftar data petugas.
Status	Valid

Tabel 6.47 Kasus Uji Mengubah Data Data Petugas

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Mengubah Data Data Petugas
Kode Kebutuhan	SAM-F-19
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “ <i>Submit</i> ”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.48 Kasus Uji Mengubah Data Data Petugas

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Mengubah Data Data Petugas
Kode Kebutuhan	SAM-F-19
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan.

	3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.49 Kasus Uji Menghapus Data Petugas

Nama Kasus Uji	Kasus Uji Menghapus Data Petugas
Kode Kebutuhan	SAM-F-20
Prosedur	1. Aktor memilih tombol "hapus". 2. Aktor memilih tombol "ok" pada pesan konfirmasi "apakah anda yakin menghapus data?". 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menghapus data petugas dan menampilkan halaman daftar data petugas.
Hasil	Sistem menghapus data petugas dan menampilkan halaman daftar data petugas.
Status	Valid

Tabel 6.50 Kasus Uji Menghapus Data Petugas

Alternatif 1: Aktor memilih tombol "cancel"

Nama Kasus Uji	Kasus Uji Menghapus Data Petugas
Kode Kebutuhan	SAM-F-20
Prosedur	1. Aktor memilih tombol "hapus". 2. Aktor memilih tombol "ok" pada pesan konfirmasi "apakah anda yakin menghapus data?". 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan data petugas dan menampilkan halaman daftar data petugas.
Hasil	Sistem membatalkan proses penghapusan data petugas dan menampilkan halaman daftar data petugas.
Status	Valid

Tabel 6.51 Kasus Uji Melihat Daftar Pengajuan Donor

Nama Kasus Uji	Kasus Uji Melihat Daftar Pengajuan Donor
Kode Kebutuhan	SAM-F-21
Prosedur	1. Aktor memilih menu petugas 2. Aktor memilih sub menu Pengajuan Donor
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar pengajuan donor berupa tabel dengan data berupa tanggal, nomor pengajuan, kode pendonor, nama pendonor, dan golongan darah.
Hasil	Sistem menampilkan halaman daftar pengajuan donor berupa tabel dengan data berupa tanggal, nomor pengajuan, kode pendonor, nama pendonor, dan golongan darah.
Status	Valid

Tabel 6.52 Kasus Uji Menambah Pengajuan Donor

Nama Kasus Uji	Kasus Uji Menambah Pengajuan Donor
Kode Kebutuhan	SAM-F-22
Prosedur	1. Aktor memilih tombol "tambah pengajuan". 2. Aktor memasukkan tanggal pengajuan, nomor pengajuan, kode pendonor pada formulir pengajuan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan tanggal pengajuan, nomor pengajuan, kode pendonor.
Hasil	Sistem menyimpan tanggal pengajuan, nomor pengajuan, kode pendonor.
Status	Valid

Tabel 6.53 Kasus Uji Menambah Pengajuan Donor**Alternatif 1: Salah satu data kosong**

Nama Kasus Uji	Kasus Uji Menambah Pengajuan Donor
Kode Kebutuhan	SAM-F-22
Prosedur	1. Aktor memilih tombol "tambah pengajuan". 2. Aktor memasukkan tanggal pengajuan dan nomor pengajuan pada formulir pengajuan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".

Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.54 Kasus Uji Menambah Pengajuan Donor

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Pengajuan Donor
Kode Kebutuhan	SAM-F-22
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah pengajuan”. 2. Aktor memasukkan tanggal pengajuan, nomor pengajuan, kode pendonor pada formulir pengajuan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.55 Kasus Uji Melihat Detail Pengajuan Donor

Nama Kasus Uji	Kasus Uji Melihat Detail Pengajuan Donor
Kode Kebutuhan	SAM-F-23
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail pengajuan donor dengan data berupa tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur pendonor, golongan darah, alamat pendonor, telepon pendonor.
Hasil	Sistem menampilkan halaman detail pengajuan donor dengan data berupa tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur pendonor, golongan darah, alamat pendonor, telepon pendonor.
Status	Valid

Tabel 6.56 Kasus Uji Menghapus Pengajuan Donor

Nama Kasus Uji	Kasus Uji Menghapus Pengajuan Donor
Kode Kebutuhan	SAM-F-24
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menghapus pengajuan donor dan menampilkan halaman daftar pengajuan donor.
Hasil	Sistem menghapus pengajuan donor dan menampilkan halaman daftar pengajuan donor.
Status	Valid

Tabel 6.57 Kasus Uji Menghapus Pengajuan Donor

Alternatif 1: Aktor memilih tombol “cancel”

Nama Kasus Uji	Kasus Uji Menghapus Pengajuan Donor
Kode Kebutuhan	SAM-F-24
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan pengajuan donor dan menampilkan halaman daftar pengajuan donor.
Hasil	Sistem membatalkan proses penghapusan pengajuan donor dan menampilkan halaman daftar pengajuan donor.
Status	Valid

Tabel 6.58 Kasus Uji Melihat Daftar Data Pendonor

Nama Kasus Uji	Kasus Uji Melihat Daftar Data Pendonor
Kode Kebutuhan	SAM-F-25
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor. 2. Aktor memilih sub menu data pendonor.
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar data pendonor berupa tabel dengan data berupa kode

	pendonor, nomor identitas, nama pendonor, umur, dan telepon.
Hasil	Sistem menampilkan halaman daftar data pendonor berupa tabel dengan data berupa kode pendonor, nomor identitas, nama pendonor, umur, dan telepon
Status	Valid

Tabel 6.59 Kasus Uji Melihat Detail Data Pendonor

Nama Kasus Uji	Kasus Uji Melihat Detail Data Pendonor
Kode Kebutuhan	SAM-F-26
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail data pendonor dengan data berupa kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.
Hasil	Sistem menampilkan halaman detail data pendonor dengan data berupa kode pendonor, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.
Status	Valid

Tabel 6.60 Kasus Uji Mengubah Data Data Pendonor

Nama Kasus Uji	Kasus Uji Mengubah Data Data Pendonor
Kode Kebutuhan	SAM-F-27
Prosedur	1. Aktor memilih tombol “edit”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan data data pendonor dan menampilkan halaman daftar data pendonor.
Hasil	Sistem akan menyimpan data data pendonor dan menampilkan halaman daftar data pendonor.
Status	Valid

Tabel 6.61 Kasus Uji Mengubah Data Data Pendoror

Alternatif 1 : Salah satu data kosong

Nama Kasus Uji	Kasus Uji Mengubah Data Data Pendoror
Kode Kebutuhan	SAM-F-27
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.62 Kasus Uji Mengubah Data Data Pendoror

Alternatif 2 : Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Mengubah Data Data Pendoror
Kode Kebutuhan	SAM-F-27
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.63 Kasus Uji Menghapus Data Pendoror

Nama Kasus Uji	Kasus Uji Menghapus Data Pendoror
Kode Kebutuhan	SAM-F-28
Prosedur	1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menghapus data pendonor dan menampilkan halaman daftar data pendonor.
Hasil	Sistem menghapus data pendonor dan menampilkan halaman daftar data pendonor.

Status	Valid
--------	-------

Tabel 6.64 Kasus Uji Menghapus Data Pendoron

Alternatif 1: Aktor memilih tombol “cancel”

Nama Kasus Uji	Kasus Uji Menghapus Data Pendoron
Kode Kebutuhan	SAM-F-28
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan Data Pendoron dan menampilkan halaman daftar Data Pendoron.
Hasil	Sistem membatalkan proses penghapusan Data Pendoron dan menampilkan halaman daftar Data Pendoron.
Status	Valid

Tabel 6.65 Kasus Uji Melihat Daftar Riwayat Donor Pendoron

Nama Kasus Uji	Kasus Uji Melihat Daftar Riwayat Donor Pendoron
Kode Kebutuhan	SAM-F-29
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor. 2. Aktor memilih sub menu riwayat donor.
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar riwayat donor berupa tabel dengan data berupa tanggal donor, kode pendonor, status kesehatan, riwayat donor, dan status riwayat penyakit.
Hasil	Sistem menampilkan halaman daftar riwayat donor berupa tabel dengan data berupa tanggal donor, kode pendonor, status kesehatan, riwayat donor, dan status riwayat penyakit.
Status	Valid

Tabel 6.66 Kasus Uji Menambah Riwayat Donor

Nama Kasus Uji	Kasus Uji Menambah Riwayat Donor
Kode Kebutuhan	SAM-F-30
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah riwayat donor”. 2. Aktor memasukkan kode pendonor, status kesehatan, riwayat donor, infeksi menular,

	riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan pada halaman formulir riwayat donor. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan.
Hasil	Sistem menyimpan kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan.
Status	Valid

Tabel 6.67 Kasus Uji Menambah Riwayat Donor

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Riwayat Donor
Kode Kebutuhan	SAM-F-30
Prosedur	1. Aktor memilih tombol "tambah riwayat donor". 2. Aktor memasukkan kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, dan kesimpulan pada halaman formulir riwayat donor. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.68 Kasus Uji Menambah Riwayat Donor Pendonor

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Riwayat Donor
Kode Kebutuhan	SAM-F-30
Prosedur	1. Aktor memilih tombol "tambah riwayat donor". 2. Aktor memasukkan kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan pada halaman formulir riwayat donor.

	3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.69 Kasus Uji Melihat Detail Riwayat Donor

Nama Kasus Uji	Kasus Uji Melihat Detail Riwayat Donor
Kode Kebutuhan	SAM-F-31
Prosedur	1. Aktor memilih tombol "lihat".
Hasil yang diharapkan	Sistem akan menampilkan halaman detail riwayat donor dengan data berupa waktu donor, kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan.
Hasil	Sistem menampilkan halaman detail riwayat donor dengan data berupa waktu donor, kode pendonor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, status riwayat penyakit, jumlah komponen, dan kesimpulan.
Status	Valid

Tabel 6.70 Kasus Uji Mengubah Data Riwayat Donor

Nama Kasus Uji	Kasus Uji Mengubah Data Riwayat Donor
Kode Kebutuhan	SAM-F-32
Prosedur	1. Aktor memilih tombol "edit". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan data riwayat donor dan menampilkan halaman daftar data riwayat donor.
Hasil	Sistem akan menyimpan data riwayat donor dan menampilkan halaman daftar riwayat donor pendonor.
Status	Valid

Tabel 6.71 Kasus Uji Mengubah Data Riwayat Donor Pendoror

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Mengubah Data Riwayat Donor Pendoror
Kode Kebutuhan	SAM-F-32
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Hasil	Sistem menampilkan pesan kesalahan validasi berupa “Bidang tersebut dibutuhkan”.
Status	Valid

Tabel 6.72 Kasus Uji Mengubah Data Riwayat Donor Pendoror

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Mengubah Data Riwayat Donor Pendoror
Kode Kebutuhan	SAM-F-32
Prosedur	1. Aktor memilih tombol “ <i>edit</i> ”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.73 Kasus Uji Menghapus Riwayat Donor

Nama Kasus Uji	Kasus Uji Menghapus Riwayat Donor
Kode Kebutuhan	SAM-F-33
Prosedur	1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menghapus riwayat donor dan menampilkan halaman daftar riwayat donor.
Hasil	Sistem menghapus riwayat donor dan menampilkan halaman daftar riwayat donor.

Status	Valid
--------	-------

Tabel 6.74 Kasus Uji Menghapus Riwayat Donor

Alternatif 1: Aktor memilih tombol “cancel”

Nama Kasus Uji	Kasus Uji Menghapus Riwayat Donor
Kode Kebutuhan	SAM-F-33
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan Riwayat Donor Pendoror dan menampilkan halaman daftar Riwayat Donor Pendoror.
Hasil	Sistem membatalkan proses penghapusan Riwayat Donor Pendoror dan menampilkan halaman daftar Riwayat Donor Pendoror.
Status	Valid

Tabel 6.75 Kasus Uji Melihat Daftar Data Butuh Darah

Nama Kasus Uji	Kasus Uji Melihat Daftar Data Butuh Darah
Kode Kebutuhan	SAM-F-34
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu pendonor. 2. Aktor memilih sub menu butuh darah.
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar butuh darah berupa tabel dengan data berupa tanggal, nama wali, nomor telepon, golongan darah, faktor rhesus(Rh), jumlah komponen, dan rumah sakit.
Hasil	Sistem menampilkan halaman daftar butuh darah berupa tabel dengan data berupa tanggal, nama wali, nomor telepon, golongan darah, faktor rhesus(Rh), jumlah komponen, dan rumah sakit.
Status	Valid

Tabel 6.76 Kasus Uji Menambah Data Butuh darah

Nama Kasus Uji	Kasus Uji Menambah Data Butuh darah
Kode Kebutuhan	SAM-F-35
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “tambah informasi”. 2. Aktor memasukkan nama wali, nomor telepon, golongan darah, faktor rhesus (rh), jumlah

	komponen, rumah sakit, keluhan, keterangan pada halaman formulir tambah data. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.
Hasil	Sistem menyimpan nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.
Status	Valid

Tabel 6.77 Kasus Uji Menambah Data Butuh darah

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Data Butuh darah
Kode Kebutuhan	SAM-F-35
Prosedur	1. Aktor memilih tombol "tambah informasi". 2. Aktor memasukkan golongan darah, faktor rhesus (rh), jumlah komponen, rumah sakit, keluhan, keterangan pada halaman formulir tambah data. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.78 Kasus Uji Menambah Data Butuh darah

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Data Butuh darah
Kode Kebutuhan	SAM-F-35
Prosedur	1. Aktor memilih tombol "tambah informasi". 2. Aktor memasukkan nama wali, nomor telepon, golongan darah, faktor rhesus (rh), jumlah komponen, rumah sakit, keluhan, keterangan pada halaman formulir tambah data. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.

Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.79 Kasus Uji Melihat Detail Butuh darah

Nama Kasus Uji	Kasus Uji Melihat Detail Butuh darah
Kode Kebutuhan	SAM-F-36
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail butuh darah dengan data berupa tanggal mengajukan, nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.
Hasil	Sistem menampilkan halaman detail butuh darah dengan data berupa tanggal mengajukan, nama wali, nomor telepon, golongan darah, faktor rhesus(rh), jumlah komponen, rumah sakit, keluhan, keterangan.
Status	Valid

Tabel 6.80 Kasus Uji Mengubah Data Butuh darah

Nama Kasus Uji	Kasus Uji Mengubah Data Butuh darah
Kode Kebutuhan	SAM-F-37
Prosedur	1. Aktor memilih tombol “edit”. 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menyimpan data butuh darah dan menampilkan halaman daftar data butuh darah.
Hasil	Sistem akan menyimpan data butuh darah dan menampilkan halaman daftar butuh darah.
Status	Valid

Tabel 6.81 Kasus Uji Mengubah Data Butuh darah

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Mengubah Data Butuh darah
Kode Kebutuhan	SAM-F-37
Prosedur	1. Aktor memilih tombol “edit”.

	2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.82 Kasus Uji Mengubah Data Butuh darah

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Mengubah Data Butuh darah
Kode Kebutuhan	SAM-F-37
Prosedur	1. Aktor memilih tombol " <i>edit</i> ". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.83 Kasus Uji Menghapus Butuh darah

Nama Kasus Uji	Kasus Uji Menghapus Butuh darah
Kode Kebutuhan	SAM-F-38
Prosedur	1. Aktor memilih tombol "hapus". 2. Aktor memilih tombol "ok" pada pesan konfirmasi "apakah anda yakin menghapus data?". 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menghapus butuh darah dan menampilkan halaman daftar butuh darah.
Hasil	Sistem menghapus butuh darah dan menampilkan halaman daftar butuh darah.
Status	Valid

Tabel 6.84 Kasus Uji Menghapus Butuh darah

Alternatif 1: Aktor memilih tombol "*cancel*"

Nama Kasus Uji	Kasus Uji Menghapus Butuh darah
-----------------------	---------------------------------

Kode Kebutuhan	SAM-F-38
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “hapus”. 2. Aktor memilih tombol “ok” pada pesan konfirmasi “apakah anda yakin menghapus data?”. 3. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan membatalkan proses penghapusan Butuh darah Pendonor dan menampilkan halaman daftar butuh darah.
Hasil	Sistem membatalkan proses penghapusan Butuh darah Pendonor dan menampilkan halaman daftar butuh darah.
Status	Valid

Tabel 6.85 Kasus Uji Melihat Daftar Data Butuh Darah

Nama Kasus Uji	Kasus Uji Melihat Daftar Data Butuh Darah
Kode Kebutuhan	SAM-F-39
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “lonceng” pemberitahuan. 2. Aktor memilih data yang akan dilihat.
Hasil yang diharapkan	Sistem akan menampilkan pemberitahuan baru yang ada.
Hasil	Sistem menampilkan pemberitahuan baru yang ada.
Status	Valid

Tabel 6.86 Kasus Uji Melihat Syarat Donor

Nama Kasus Uji	Kasus Uji Melihat Syarat Donor
Kode Kebutuhan	SAM-F-40
Prosedur	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem.
Hasil yang diharapkan	Sistem akan menampilkan poin-poin yang menjelaskan syarat-syarat donor.
Hasil	Sistem menampilkan poin-poin yang menjelaskan syarat-syarat donor.
Status	Valid

Tabel 6.87 Kasus Uji Menambah Pengajuan Donor

Nama Kasus Uji	Kasus Uji Menambah Pengajuan Donor
Kode Kebutuhan	SAM-F-41
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Pengajuan Donor”. 2. Aktor mengecek kebenaran data.

	<p>3. Aktor memasukkan tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, telepon pendonor, tempat lahir, tanggal lahir, golongan darah, umur pendonor, alamat pendonor, dan email pada formulir pengajuan.</p> <p>4. Aktor memilih tombol "Submit".</p>
Hasil yang diharapkan	Sistem akan menyimpan tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, telepon pendonor, tempat lahir, tanggal lahir, golongan darah, umur pendonor, alamat pendonor, dan email.
Hasil	Sistem menyimpan tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, telepon pendonor, tempat lahir, tanggal lahir, golongan darah, umur pendonor, alamat pendonor, dan email.
Status	Valid

Tabel 6.88 Kasus Uji Menambah Data Butuh darah
Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Pengajuan Donor
Kode Kebutuhan	SAM-F-41
Prosedur	<p>1. Aktor memilih tombol "Pengajuan Donor".</p> <p>2. Aktor mengecek kebenaran data.</p> <p>3. Aktor memasukkan tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, telepon pendonor, tempat lahir, tanggal lahir, golongan darah, umur pendonor, alamat pendonor, dan email pada formulir pengajuan.</p> <p>4. Aktor memilih tombol "Submit".</p>
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.89 Kasus Uji Menambah Data Butuh darah
Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Pengajuan Donor
Kode Kebutuhan	SAM-F-41
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Pengajuan Donor”. 2. Aktor mengecek kebenaran data. 3. Aktor memasukkan tanggal pengajuan, nomor pengajuan, kode pendonor, nomor identitas, nama pendonor, telepon pendonor, tempat lahir, tanggal lahir, golongan darah, umur pendonor, alamat pendonor, dan email pada formulir pengajuan. 4. Aktor memilih tombol “Submit”.
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.90 Kasus Uji Mencetak Pengajuan Donor

Nama Kasus Uji	Kasus Uji Mencetak Pengajuan Donor
Kode Kebutuhan	SAM-F-42
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih tombol “print”. 2. Aktor memilih tombol “print” pada format formulir pengajuan donor.
Hasil yang diharapkan	Sistem akan mencetak pengajuan donor.
Hasil	Sistem mencetak pengajuan donor.
Status	Valid

Tabel 6.91 Kasus Uji Melihat Informasi Butuh Darah

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan halaman informasi butuh donor darah sesuai dengan golongan darah aktor yang telah terotorisasi masuk ke dalam sistem
Hasil	Sistem menampilkan halaman informasi butuh donor darah sesuai dengan golongan darah aktor yang telah terotorisasi masuk ke dalam sistem.

Status	Valid
--------	-------

Tabel 6.92 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 1: golongan darah pendonor yang telah terotorisasi bergolongan darah O+

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah O+, A+, B+, dan AB+ pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah O+, A+, B+, dan AB+ pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.93 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 2: golongan darah pendonor yang telah terotorisasi bergolongan darah A+

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah A+ dan AB+ pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah A+ dan AB+ pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.94 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 3: golongan darah pendonor yang telah terotorisasi bergolongan darah B+

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah B+ dan AB+ pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah B+ dan AB+ pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.95 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 4: golongan darah pendonor yang telah terotorisasi bergolongan darah AB+

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah AB+ pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah AB+ pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.96 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 5: golongan darah pendonor yang telah terotorisasi bergolongan darah O-

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah O+, A+, B+, AB+, O-, A-, B-, dan AB- pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah O+, A+, B+, AB+, O-, A-, B-, dan AB- pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.97 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 6: golongan darah pendonor yang telah terotorisasi bergolongan darah A-

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah A+, AB+, A- dan AB- pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah A+, AB+, A- dan AB- pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.98 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 7: golongan darah pendonor yang telah terotorisasi bergolongan darah B-

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah B+, AB+, B- dan AB- pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah B+, AB+, B- dan AB- pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.99 Kasus Uji Melihat Informasi Butuh Darah

Alternatif 8: golongan darah pendonor yang telah terotorisasi bergolongan darah AB-

Nama Kasus Uji	Kasus Uji Melihat Informasi Butuh Darah
Kode Kebutuhan	SAM-F-43
Prosedur	1. Aktor memilih menu butuh donor darah.
Hasil yang diharapkan	Sistem akan menampilkan data golongan darah AB+ dan AB- pada halaman informasi butuh donor darah.
Hasil	Sistem menampilkan data golongan darah AB+ dan AB- pada halaman informasi butuh donor darah.
Status	Valid

Tabel 6.100 Kasus Uji Melihat Profil Pendonor

Nama Kasus Uji	Kasus Uji Melihat Profil Pendonor
Kode Kebutuhan	SAM-F-44
Prosedur	1. Aktor memilih menu " <i>setting</i> profil".
Hasil yang diharapkan	Sistem akan menampilkan halaman <i>setting</i> profil pada tab profil yang berisi kode pendonor, username, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.
Hasil	Sistem menampilkan halaman <i>setting</i> profil pada tab profil yang berisi kode pendonor, username, nomor identitas, nama pendonor, tempat lahir, tanggal lahir, umur, golongan darah, alamat, telepon, dan email.
Status	Valid

Tabel 6.101 Kasus Uji Mengubah Profil Pendoron

Nama Kasus Uji	Kasus Uji Mengubah Profil Pendoron
Kode Kebutuhan	SAM-F-45
Prosedur	1. Aktor memilih tombol <i>"edit"</i> . 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol <i>"Submit"</i> .
Hasil yang diharapkan	Sistem akan menyimpan profil pendonor dan menampilkan halaman setting profil.
Hasil	Sistem menyimpan profil pendonor dan menampilkan halaman setting profil.
Status	Valid

Tabel 6.102 Kasus Uji Mengubah Profil Pendoron**Alternatif 1: Salah satu data kosong**

Nama Kasus Uji	Kasus Uji Mengubah Profil Pendoron
Kode Kebutuhan	SAM-F-45
Prosedur	1. Aktor memilih tombol <i>"edit"</i> . 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol <i>"Submit"</i> .
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa <i>"Bidang tersebut dibutuhkan"</i> .
Hasil	Sistem menampilkan pesan kesalahan validasi berupa <i>"Bidang tersebut dibutuhkan"</i> .
Status	Valid

Tabel 6.103 Kasus Uji Mengubah Profil Pendoron**Alternatif 2: Salah satu data belum terisi dengan benar**

Nama Kasus Uji	Kasus Uji Mengubah Profil Pendoron
Kode Kebutuhan	SAM-F-45
Prosedur	1. Aktor memilih tombol <i>"edit"</i> . 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol <i>"Submit"</i> .
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.104 Kasus Uji Melihat Penghargaan Donor

Nama Kasus Uji	Kasus Uji Melihat Penghargaan Donor
Kode Kebutuhan	SAM-F-46
Prosedur	1. Aktor memilih menu “Riwayat Donor”.
Hasil yang diharapkan	Sistem akan menampilkan halaman riwayat donor dan menampilkan informasi penghargaan donor yang berisi alur penghargaan dan jenis penghargaan yang akan diterima dari jumlah riwayat donor pendonor.
Hasil	Sistem menampilkan halaman riwayat donor dan menampilkan informasi penghargaan donor yang berisi alur penghargaan dan jenis penghargaan yang akan diterima dari jumlah riwayat donor pendonor.
Status	Valid

Tabel 6.105 Kasus Uji Melihat Riwayat Donor

Nama Kasus Uji	Kasus Uji Melihat Riwayat Donor
Kode Kebutuhan	SAM-F-47
Prosedur	1. Aktor memilih menu “Riwayat Donor”.
Hasil yang diharapkan	Sistem akan menampilkan halaman riwayat donor dan menampilkan tabel informasi riwayat donor yang berisi tanggal donor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, riwayat penyakit, jumlah donor dan kesimpulan.
Hasil	Sistem menampilkan halaman riwayat donor dan menampilkan tabel informasi riwayat donor yang berisi tanggal donor, status kesehatan, riwayat donor, infeksi menular, riwayat tinggal, riwayat penyakit, jumlah donor dan kesimpulan.
Status	Valid

Tabel 6.106 Kasus Uji Menambah Pengajuan permintaan darah

Nama Kasus Uji	Kasus Uji Menambah Pengajuan permintaan darah
Kode Kebutuhan	SAM-F-48
Prosedur	1. Aktor memilih tombol “tambah permintaan”. 2. Aktor memasukkan nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis

	kelamin, golongan darah dan file bukti surat dokter pada halaman formulir permintaan darah. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menyimpan nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, golongan darah dan file bukti surat dokter.
Hasil	Sistem menyimpan nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, golongan darah dan file bukti surat dokter.
Status	Valid

Tabel 6.107 Kasus Uji Menambah Pengajuan permintaan darah

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah Pengajuan permintaan darah
Kode Kebutuhan	SAM-F-48
Prosedur	1. Aktor memilih tombol "tambah permintaan". 2. Aktor memasukkan nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, golongan darah dan file bukti surat dokter pada halaman formulir permintaan darah. 3. Aktor memilih tombol "Submit".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.108 Kasus Uji Menambah Pengajuan permintaan darah

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Menambah Pengajuan permintaan darah
Kode Kebutuhan	SAM-F-48
Prosedur	1. Aktor memilih tombol "tambah permintaan".

	<p>2. Aktor memasukkan nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, golongan darah dan file bukti surat dokter pada halaman formulir permintaan darah.</p> <p>3. Aktor memilih tombol "Submit".</p>
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

Tabel 6.109 Kasus Uji Menambah Pengajuan pengajuan permintaan darah

Alternatif 3: Upload file gagal atau tidak sesuai dengan peraturan

Nama Kasus Uji	Kasus Uji Menambah Pengajuan pengajuan permintaan darah
Kode Kebutuhan	SAM-F-48
Prosedur	<p>1. Aktor memilih tombol "tambah permintaan".</p> <p>2. Aktor memasukkan nomor permintaan, kode petugas, nama rumah sakit, nama petugas, alamat rumah sakit, telepon rumah sakit, nama dokter, jumlah komponen, jenis komponen, nama pasien, nomor med.rec, umur, agama, jenis kelamin, golongan darah dan file bukti surat dokter pada halaman formulir permintaan darah.</p> <p>3. Aktor memilih tombol "Submit".</p>
Hasil yang diharapkan	Sistem akan menampilkan pesan error.
Hasil	Sistem menampilkan pesan error.
Status	Valid

Tabel 6.110 Kasus Uji Melihat Daftar Pengajuan Permintaan Darah

Nama Kasus Uji	Kasus Uji Melihat Daftar Pengajuan Permintaan Darah
Kode Kebutuhan	SAM-F-49
Prosedur	<p>1. Aktor memilih menu permintaan darah.</p> <p>2. Aktor memilih tab menu menunggu persetujuan.</p>

Hasil yang diharapkan	Sistem akan menampilkan daftar permintaan darah pada tab menu menunggu persetujuan berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Hasil	Sistem menampilkan daftar permintaan darah pada tab menu menunggu persetujuan berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Status	Valid

Tabel 6.111 Kasus Uji Melihat Detail Pengajuan Permintaan Darah

Nama Kasus Uji	Kasus Uji Melihat Detail Pengajuan Permintaan Darah
Kode Kebutuhan	SAM-F-50
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail permintaan darah pada tab menu menunggu persetujuan yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Hasil	Sistem menampilkan halaman detail permintaan darah pada tab menu menunggu persetujuan yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Status	Valid

Tabel 6.112 Kasus Uji Mencetak Pengajuan Permintaan Darah

Nama Kasus Uji	Kasus Uji Mencetak Pengajuan Permintaan Darah
Kode Kebutuhan	SAM-F-51
Prosedur	1. Aktor memilih tombol “cetak”. 2. Aktor memilih tombol “print” pada format formulir permintaan darah.
Hasil yang diharapkan	Sistem akan mencetak permintaan darah.
Hasil	Sistem mencetak permintaan darah.
Status	Valid

Tabel 6.113 Kasus Uji Melihat Daftar Permintaan disetujui

Nama Kasus Uji	Kasus Uji Melihat Daftar Permintaan disetujui
Kode Kebutuhan	SAM-F-52
Prosedur	1. Aktor memilih tab menu permintaan disetujui.
Hasil yang diharapkan	Sistem akan menampilkan daftar permintaan darah pada tab menu permintaan disetujui berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Hasil	Sistem menampilkan daftar permintaan darah pada tab menu permintaan disetujui berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Status	Valid

Tabel 6.114 Kasus Uji Melihat Detail Permintaan disetujui

Nama Kasus Uji	Kasus Uji Melihat Detail Permintaan disetujui
Kode Kebutuhan	SAM-F-53
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail permintaan darah pada tab menu permintaan disetujui yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Hasil	Sistem menampilkan halaman detail permintaan darah pada tab menu permintaan disetujui yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Status	Valid

Tabel 6.115 Kasus Uji Melihat Daftar Darah diterima

Nama Kasus Uji	Kasus Uji Melihat Daftar Darah diterima
Kode Kebutuhan	SAM-F-54
Prosedur	1. Aktor memilih tab menu darah diterima.
Hasil yang diharapkan	Sistem akan menampilkan daftar permintaan darah pada tab menu darah diterima berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah

	komponen, jenis komponen, nama pasien, dan nomor med.rec.
Hasil	Sistem menampilkan daftar permintaan darah pada tab menu darah diterima berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Status	Valid

Tabel 6.116 Kasus Uji Melihat Detail Darah diterima

Nama Kasus Uji	Kasus Uji Melihat Detail Darah diterima
Kode Kebutuhan	SAM-F-55
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail permintaan darah pada tab menu darah diterima yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Hasil	Sistem menampilkan halaman detail permintaan darah pada tab menu darah diterima yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Status	Valid

Tabel 6.117 Kasus Uji Melihat Daftar Permintaan Selesai

Nama Kasus Uji	Kasus Uji Melihat Daftar Permintaan Selesai
Kode Kebutuhan	SAM-F-56
Prosedur	1. Aktor memilih tab menu pesan selesai.
Hasil yang diharapkan	Sistem akan menampilkan daftar permintaan darah pada tab menu pesan selesai berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Hasil	Sistem menampilkan daftar permintaan darah pada tab menu pesan selesai berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Status	Valid

Tabel 6.118 Kasus Uji Melihat Detail Permintaan Selesai

Nama Kasus Uji	Kasus Uji Melihat Detail Permintaan Selesai
Kode Kebutuhan	SAM-F-57
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail permintaan darah pada tab menu pesan selesai yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Hasil	Sistem menampilkan halaman detail permintaan darah pada tab menu pesan selesai yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Status	Valid

Tabel 6.119 Kasus Uji Mencetak Permintaan Selesai

Nama Kasus Uji	Kasus Uji Mencetak Permintaan Selesai
Kode Kebutuhan	SAM-F-58
Prosedur	1. Aktor memilih tombol “cetak”. 2. Aktor memilih tombol “print” pada format formulir permintaan darah.
Hasil yang diharapkan	Sistem akan mencetak permintaan selesai.
Hasil	Sistem mencetak permintaan selesai.
Status	Valid

Tabel 6.120 Kasus Uji Melihat Daftar Permintaan Dibatalkan

Nama Kasus Uji	Kasus Uji Melihat Daftar Permintaan Dibatalkan
Kode Kebutuhan	SAM-F-59
Prosedur	1. Aktor memilih tab menu pesan batal.
Hasil yang diharapkan	Sistem akan menampilkan daftar permintaan darah pada tab menu pesan batal berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.
Hasil	Sistem menampilkan daftar permintaan darah pada tab menu pesan batal berupa tabel dengan data berisi tanggal, status permintaan, nama dokter, nomor permintaan, golongan darah, jumlah komponen, jenis komponen, nama pasien, dan nomor med.rec.

Status	Valid
---------------	-------

Tabel 6.121 Kasus Uji Melihat Detail Permintaan Dibatalkan

Nama Kasus Uji	Kasus Uji Melihat Detail Permintaan Dibatalkan
Kode Kebutuhan	SAM-F-60
Prosedur	1. Aktor memilih tombol “lihat”.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail permintaan darah pada tab menu pesan batal yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Hasil	Sistem menampilkan halaman detail permintaan darah pada tab menu pesan batal yang berisi alur permintaan darah, alur pengiriman darah, dan detail permintaan darah.
Status	Valid

Tabel 6.122 Kasus Uji Mencetak Permintaan Dibatalkan

Nama Kasus Uji	Kasus Uji Mencetak Permintaan Dibatalkan
Kode Kebutuhan	SAM-F-61
Prosedur	1. Aktor memilih tombol “cetak”. 2. Aktor memilih tombol “print” pada format formulir permintaan darah.
Hasil yang diharapkan	Sistem akan mencetak permintaan dibatalkan.
Hasil	Sistem mencetak permintaan dibatalkan.
Status	Valid

Tabel 6.123 Kasus Uji Melihat Syarat Permintaan Darah

Nama Kasus Uji	Kasus Uji Melihat Syarat Permintaan Darah
Kode Kebutuhan	SAM-F-62
Prosedur	1. Aktor memilih menu informasi.
Hasil yang diharapkan	Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan informasi poin-poin yang menjelaskan syarat permintaan darah.
Hasil	Sistem menampilkan halaman informasi. Halaman tersebut menampilkan informasi poin-poin yang menjelaskan syarat permintaan darah.
Status	Valid

Tabel 6.124 Kasus Uji Melihat Informasi Alur Permintaan

Nama Kasus Uji	Kasus Uji Melihat Informasi Alur Permintaan
Kode Kebutuhan	SAM-F-63
Prosedur	1. Aktor memilih menu informasi.
Hasil yang diharapkan	Sistem akan menampilkan halaman informasi. Halaman tersebut menampilkan informasi poin-poin yang menjelaskan alur permintaan.
Hasil	Sistem menampilkan halaman informasi. Halaman tersebut menampilkan informasi poin-poin yang menjelaskan alur permintaan.
Status	Valid

Tabel 6.125 Kasus Uji Melihat Profil Petugas Rumah Sakit

Nama Kasus Uji	Kasus Uji Melihat Profil Petugas Rumah Sakit
Kode Kebutuhan	SAM-F-64
Prosedur	1. Aktor memilih tombol " <i>edit profil</i> " pada <i>dropdown</i> nama petugas rumah sakit.
Hasil yang diharapkan	Sistem akan menampilkan halaman detail profil yang berisi kode petugas, username, nama petugas, nama rumah sakit, alamat rumah sakit, dan telepon rumah sakit.
Hasil	Sistem menampilkan halaman detail profil yang berisi kode petugas, username, nama petugas, nama rumah sakit, alamat rumah sakit, dan telepon rumah sakit.
Status	Valid

Tabel 6.126 Kasus Uji Mengubah Profil Petugas Rumah Sakit

Nama Kasus Uji	Kasus Uji Mengubah Profil Petugas Rumah Sakit
Kode Kebutuhan	SAM-F-65
Prosedur	1. Aktor memilih tombol " <i>Edit profil</i> ". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol " <i>Submit</i> ".
Hasil yang diharapkan	Sistem akan menyimpan profil petugas rumah sakit dan menampilkan halaman setting profil.
Hasil	Sistem menyimpan profil petugas rumah sakit dan menampilkan halaman setting profil.
Status	Valid

Tabel 6.127 Kasus Uji Mengubah Profil Petugas Rumah Sakit

Alternatif 1: Salah satu data kosong

Nama Kasus Uji	Kasus Uji Mengubah Profil Petugas Rumah Sakit
Kode Kebutuhan	SAM-F-65
Prosedur	1. Aktor memilih tombol " <i>Edit</i> profil". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol " <i>Submit</i> ".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Hasil	Sistem menampilkan pesan kesalahan validasi berupa "Bidang tersebut dibutuhkan".
Status	Valid

Tabel 6.128 Kasus Uji Mengubah Profil Petugas Rumah Sakit

Alternatif 2: Salah satu data belum terisi dengan benar

Nama Kasus Uji	Kasus Uji Mengubah Profil Petugas Rumah Sakit
Kode Kebutuhan	SAM-F-65
Prosedur	1. Aktor memilih tombol " <i>Edit</i> profil". 2. Aktor mengubah data sesuai yang diinginkan. 3. Aktor memilih tombol " <i>Submit</i> ".
Hasil yang diharapkan	Sistem akan menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Hasil	Sistem menampilkan pesan kesalahan validasi berdasarkan peraturan validasi yang telah ditentukan.
Status	Valid

6.3 Pengujian *Compatibility*

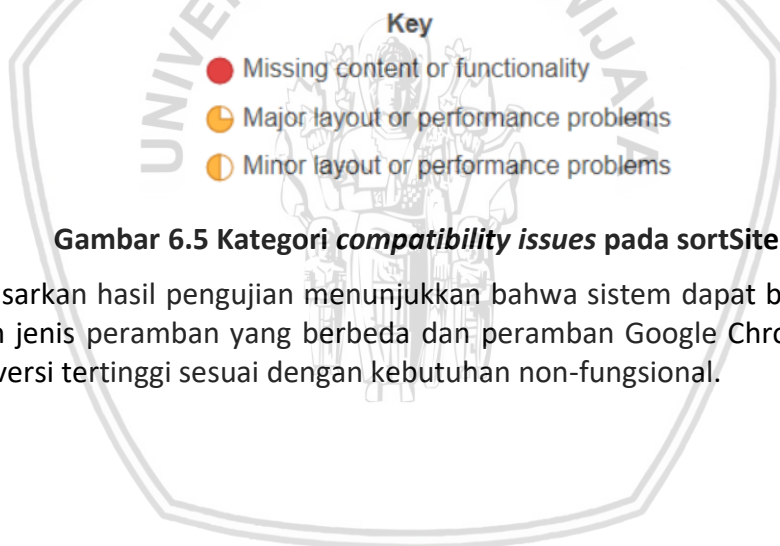
Pengujian *compatibility* yang diterapkan pada sistem ini dilakukan berdasarkan pada sisi perambah web. Pengujian *compatibility* dilakukan dengan menggunakan perangkat lunak bernama SortSite versi 5, alat ini menganalisa struktur *website* untuk mengetahui apakah terdapat bagian yang tidak kompatibel dengan perambah web.

Browser Version	IE 9 10 11	Edge 16	Firefox 60	Safari ≤ 10 11	Opera 51	Chrome 66	iOS ≤ 9 10 11	Android ≤ 3 4* ≤ 7.1 10.0	BlackBerry
Critical Issues	✓ ✓ ✓	✓	✓	✓ ✓	✓	✓	✓ ✓ ✓	✓ ✓	✓
Major Issues	✓ ✓ ✓	✓	✓	✓ ✓	✓	✓	✓ ✓ ✓	✓ ✓	✓
Minor Issues	✓ ✓ ✓	✓	✓	✓ ✓	✓	✓	✓ ✓ ✓	✓ ✓	✓

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 6.4 Hasil Pengujian *Compatibility* Aplikasi

Berdasarkan Gambar 6.4, dapat terlihat masalah apa yang terdapat dalam sistem berdasarkan tiga kategori yaitu, *critical issues*, *major issues*, dan *minor issues*. *Critical issues* mengindikasikan sebuah konten atau fitur yang tidak didukung pada beberapa peramban web. *Major issues* mengindikasikan masalah utama pada tampilan atau performa pada beberapa peramban web. *Minor issues* mengindikasikan masalah minor pada tampilan atau performa pada beberapa peramban web. Gambar 6.5 merupakan penjelasan indikator kategori masalah *compatibility* pada sortSite.



Gambar 6.5 Kategori *compatibility issues* pada sortSite

Berdasarkan hasil pengujian menunjukkan bahwa sistem dapat berjalan pada sembilan jenis peramban yang berbeda dan peramban Google Chrome versi 66 sebagai versi tertinggi sesuai dengan kebutuhan non-fungsional.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan dari hasil pengamatan analisis kebutuhan, perancangan, implementasi, dan pengujian, maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan, sistem aplikasi manajemen Unit Transfusi Darah (UTD) PMI Malang memiliki 4 aktor yang terlibat dalam sistem, yaitu guest, administrator, pendonor, dan petugas rumah sakit. Dalam proses analisis kebutuhan juga menghasilkan 65 kebutuhan fungsional dan 1 kebutuhan non fungsional. Use case diagram dan use case scenario dalam analisis kebutuhan termasuk ke dalam pemodelan kebutuhan sistem dengan tujuan use case diagram digunakan untuk mengetahui interaksi apa saja yang ada antara aktor dengan sistem yang memiliki 65 use case. Dan untuk use case scenario bertujuan untuk menjabarkan alur kinerja pada setiap use case diagram sistem.
2. Berdasarkan perancangan yang telah dilakukan, maka dihasilkan 3 sequence diagram dari fitur utama yaitu menambah alur pengiriman darah, melihat informasi butuh darah, dan menambah permintaan darah. Perancangan basis data yang digunakan dalam sistem adalah menggunakan *Physical Data Model* (PDM) yang menghasilkan 9 tabel, yaitu riwayatdonor, users, akunpetugas, saran, profilpendonor, formpermintaan, butuh, daftarpengajuan, dan alurpengiriman. Sedangkan untuk *class diagram* pada proses perancangan menghasilkan 12 class pada *controller* dan 4 class pada model. Perancangan algoritme menghasilkan *pseudocode* yang akan digunakan dalam proses implementasi. Sedangkan perancangan antarmuka menjelaskan tentang contoh baku perancangan antarmuka yang akan diterapkan kedalam sistem.
3. Berdasarkan implementasi yang telah dilakukan, maka dihasilkan spesifikasi sistem untuk melakukan implementasi dengan menggunakan perangkat keras dan perangkat lunak. Implementasi algoritme diperoleh dari perancangan algoritme dan diimplementasikan menjadi kode program. Implementasi antarmuka juga diperoleh dari perancangan antarmuka yang telah dirancang sebelumnya.
4. Berdasarkan pengujian unit, pengujian validasi, dan pengujian compatibility yang telah dilakukan, maka dihasilkan pengujian unit menambah alur pengiriman darah memiliki nilai *cyclomatic complexity* 6 yang terdapat 6 jalur independen, sedangkan untuk pengujian unit melihat informasi butuh darah memiliki nilai *cyclomatic complexity* 8 dan menambah permintaan darah memiliki nilai *cyclomatic complexity* 4. Dari ketiga pengujian unit yang telah dilakukan dapat diketahui bahwa sistem manajemen Unit Transfusi Darah (UTD) dinyatakan lolos pengujian unit, karena menghasilkan nilai 100% valid pada semua jalur uji. Untuk pengujian validasi diketahui bahwa terdapat 120 kasus uji yang telah didefinisikan berstatus 100% valid. Hal tersebut berarti seluruh spesifikasi kebutuhan perangkat lunak telah terpenuhi, oleh karena

itu sistem manajemen Unit Transfusi Darah (UTD) dinyatakan lolos uji validasi. Dan untuk pengujian *compatibility* membagi masalah menjadi tiga jenis yaitu *critical issues*, *major issues*, dan *minor issues*. Pada pengujian *compatibility* yang dilakukan terhadap 9 jenis peramban web dapat diketahui bahwa tidak terdapat *critical issues*, *major issues*, dan *minor issues*. Hal tersebut menunjukkan bahwa perangkat lunak dapat berjalan dengan baik di semua peramban yang diuji tanpa ada masalah yang berarti.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem aplikasi manajemen Unit Transfusi Darah (UTD) PMI Malang selanjutnya antara lain:

1. Sistem dapat dikembangkan untuk memiliki fitur pemberitahuan untuk pengajuan donor terhadap petugas rumah sakit, batas waktu masa berlaku formulir pengajuan donor dari pendonor mengajukan hingga pendonor akan mendonor, dan pemberitahuan melalui SMS (*Short Message Service*) apabila akan mendekati hari terakhir masa berlaku pengajuan donor,
2. Sistem dapat dikembangkan untuk memiliki fitur pembayaran kantong darah setelah proses permintaan darah telah selesai, laporan permintaan darah berdasarkan bulan hingga tahun.
3. Sistem dapat dikembangkan untuk memiliki fitur mengelola darah masuk dan darah keluar setiap bulannya, memiliki fitur untuk peminjaman mobil unit donor darah, serta perizinan online yang dapat dilakukan secara efektif.

DAFTAR PUSTAKA

- Atmosudirdjo, S. Prajudi. 1986. *Dasar-dasar Ilmu Administrasi*. Jakarta: Ghalia Indonesia.
- Blanco, J.A. dan Upton, D. 2009. *Codeigniter 1.7: Improve your PHP coding productivity with the free compact open source MVC Codeigniter framework!*. Birmingham, U.K: Pack Publishing.
- Bassil, Y. 2012. *A Simulation Model For The Waterfall Software Development Life Cycle*. International Journal of Engineering & Technology (iJET). Vol.2, No.5, ISSN:2049-3444, Beirut: Lebanese Association for Computational Sciences.
- Booch, Grady & et all. 2005. *The Unified Modeling Language User Guide Second Edition*. United States: Addison-Wesley.
- Chopra, Rajiv. 2012. *Reusing Black Box Test Paths for White Box Testing of Websites*. IEEE International Advance Computing Conference (IACC). ISBN: 978-1-4673-4529-3.
- Mader, Sylvia S. 2012. *Biology 11th Edition*. London: McGraw-Hill Higher Education.
- Manullang, M. 2005. *Dasar dasar Manajemen*. Yogyakarta: Gadjah Mada University Press.
- Pemenkes RI. 2015. *Undang-Undang No. 91 Tahun 2015 tentang standar pelayanan transfusi darah*. Jakarta: Pemenkes RI.
- Pressman, Roger S. 2001. *Software Engineering: A Practitioner's Approach, Fifth Edition*. New York: McGraw-Hill.
- Pressman, Roger S. 2010. *Software Engineering: A Practitioner's Approach, 7th Edition*. New York: McGraw-Hill.
- Rizky, Soetam. 2011. *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta: Prestasi Pustaka.
- Rosa A.S. dan Salahuddin M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Siagian, P. Sondang. 2005. *Sistem Informasi Manajemen*. Jakarta: Bumi Aksara.
- Sidik, B. 2012. *Framework Codeigniter*. Bandung: Penerbit Informatika.
- Sofwan, A. 2012. *Belajar PHP dengan Framework Codeigniter*. Tersedia di: <<http://ilmukomputer.org/2010/05/02/belajar-php-dengan-framework-code-igniter/>> [Diakses 19 April 2018]
- Sommerville, Ian. 2011. *Software Engineering 9th edition*. Pearson Education, Inc.
- Upton, D. 2007. *Codeigniter for Rapid PHP Application Development*. Birmingham, U.K: Pack Publishing.

Widodo. 2012. *Psikologi Belajar*. Jakarta: Rineka Cipta.

Whitten, Jeffery L., Lonnie D. Bentley, Kevin C. Dittman. 2004. *Systems Analysis and Design Methods*. 6 edition. McGraw-Hill.

